SoK: Security of the Ascon Modes

Charlotte Lefevre and Bart Mennink

Digital Security Group, Radboud University, Nijmegen, The Netherlands
charlotte.lefevre@ru.nl, b.mennink@cs.ru.nl

Abstract. The Ascon authenticated encryption scheme and hash function of Dobraunig et al. (Journal of Cryptology 2021) were recently selected as winner of the NIST lightweight cryptography competition. The mode underlying Ascon authenticated encryption (Ascon-AE) resembles ideas of SpongeWrap, but not quite, and various works have investigated the generic security of Ascon-AE, all covering different attack scenarios and with different bounds. This work systematizes knowledge on the mode security of Ascon-AE, and fills gaps where needed. We consider six mainstream security models, all in the multi-user setting: (i) nonce-respecting security, reflecting on the existing bounds of Chakraborty et al. (ASIACRYPT 2023, ACISP 2024) and Lefevre and Mennink (SAC 2024), (ii) nonce-misuse resistance, observing a non-fixable flaw in the proof of Chakraborty et al. (ACISP 2024), (iii) nonce-misuse resilience, delivering missing security analysis, (iv) leakage resilience, delivering a new security analysis that supersedes the informal proof sketch (though in a different model) of Guo et al. (ToSC 2020), (v) state-recovery security, expanding on the analysis of Lefevre and Mennink, and (vi) release of unverified plaintext, also delivering missing security analysis. We also match all bounds with tight attacks (up to constant and up to reasonable assumptions). As a bonus, we systematize the knowledge on Ascon-Hash and Ascon-PRF.

Keywords: Ascon · lightweight cryptography · mode security · SoK

1 Introduction

For the last few decades, lightweight cryptography has been a dominant research domain in the field of symmetric cryptography. The field of lightweight cryptography concentrates on the design of cryptographic schemes that would be efficient and secure even if certain constraints on, e.g., area, power consumption, or latency are in place. Perhaps one of the earliest proposal that fits this description is the NOEKEON block cipher [DPVR00], dating back to 2000, but the field only started to gain traction a few years later, with the introduction of explicitly lightweight branded block ciphers HIGHT [HSH+06], PRESENT [BKL+07], and KATAN [CDK09] in the late 2000s. A large body of lightweight (tweakable) block ciphers has been published since, including (but not limited to) PRINCE [BCG+12], SIMON [BSS+13], SKINNY [BJK+16], GIFT [BPP+17], QARMA [Ava17], and QARMAv2 [ABD+23].

The rise of lightweight cryptography happened together with the introduction of modern permutation-based cryptography and cryptographic sponge functions in 2007 [BDPV07, BDPV08, NIS07, Nat15]. In a nutshell, a sponge operates on top of a b-bit permutation p and maintains a b-bit state S that is split into a c-bit inner part (where c is called the capacity) and an r-bit outer part (where r is called the rate). It then absorbs a plaintext P by injectively padding it into r-bit blocks $(P_1, \ldots, P_v)$, and bitwise adds those blocks to the outer part of the state, interleaved with an evaluation of p: $S \leftarrow p(S \oplus P_i \| 0^c)$. After the last plaintext block is absorbed, the sponge squeezes r bits at a time from the outer part, namely $\lceil S \rceil_r$, again interleaved with evaluations of p, $S \leftarrow p(S)$. We refer

to Figure 4 in Section 8 for a visualization of this mode. It was quickly acknowledged that sponge functions are extremely well-suited for the design of lightweight cryptography. Indeed, the early lightweight hash functions QUARK [AHMN10], PHOTON [GPP11], and SPONGENT [BKL+11] are, in fact, sponge functions. Likewise, the keyed sibling of the sponge, the duplex [BDPV11a, DMV17], turned out to be very well-suited for the design of lightweight authenticated encryption, and fundamental research as well as the development of designs in this direction has been significantly boosted by two competitions: the CAESAR competition for authenticated encryption design [CAE14] and thereafter by the lightweight cryptography competition organized by the US National Institute of Standards and Technology (NIST) [NIS19]. In the CAESAR competition, there were 10 out of 57 submissions based on or inspired by the duplex, and eventually, the Ascon authenticated encryption scheme [DEMS21,DEMS14] was selected as winner in the category lightweight. In the NIST lightweight cryptography competition, 22 out of 57 submissions were duplex-inspired, and Ascon [DEMS21, DEMS19] was even selected as overall winner. This means that Ascon will soon be standardized as lightweight authenticated encryption scheme, with a draft of the standard already open to public comments [SMKK24].

In a bit more detail, Ascon typically refers to the authenticated encryption scheme, dubbed Ascon-AE in this work to avoid ambiguity. (Looking ahead, the upcoming NIST standard also includes a hash function standard Ascon-Hash, a XOF Ascon-XOF, and a customized XOF Ascon-CXOF, and another relevant scheme is a PRF called Ascon-PRF [DEMS24], but we will come back to these later.) Ascon-AE is an authenticated encryption scheme inspired by the duplex construction [BDPV11a, MRV15, DMV17], but with some subtle differences. In a nutshell, Ascon-AE operates on two ($b = 320$)-bit permutations, an outer permutation p_o and inner permutation p_i, which differ in the number of rounds and round constants. For a new authenticated encryption operation, it initializes a 320-bit state with a 64-bit initialization vector (encoding the specific instance), a 128-bit key, and a 128-bit nonce. Then, it permutes the state using the outer permutation p_o, and compresses the key again into the state. Then, the scheme processes the associated data by absorbing it block-by-block into the state, interleaved with evaluations of p_i, and encrypts the plaintext block-by-block by using part of the state as keystream and subsequently absorbing the plaintext into the state, again interleaved with evaluations of p_i. Finally, the state is blinded once again with the key, a last evaluation of p_o is made and a t-bit chunk of the state is blinded a final time with the key before it is output as tag. We refer to Section 2 for a detailed description of the Ascon-AE mode, including a visualization in Figure 2a. Sometimes, when looking at the construction generically, we discard the difference between the two permutations and assume a single permutation p.

As mentioned, the upcoming NIST standard [SMKK24] also includes a hash function standard Ascon-Hash, a XOF Ascon-XOF, and a customized XOF Ascon-CXOF. They are fairly direct instantiations of the aforementioned sponge, operating on the 320-bit permutation p. The difference between the hash function and the XOFs is that Ascon-Hash only outputs fixed-length digests but Ascon-XOF/Ascon-CXOF accommodate for variable-length digests. The difference between the two XOFs is that the customized XOF allows for a customization string to be prepended to the plaintext. Refer to Section 8 for a specification of Ascon-Hash, Ascon-XOF, and Ascon-CXOF. Another scheme worth mentioning is a PRF called Ascon-PRF [DEMS24], of which the main goal is to authenticate the plaintext. Ascon-PRF is basically a keyed sponge construction. It can be seen to operate the Ascon-XOF function but initialized with an initialization vector and a 128-bit key, and right before squeezing the tag, domain separation is applied by flipping a bit in the inner part. A description of Ascon-PRF is given in Section 9.

1.1 Generic Security of Sponges and Duplexes

The sponge/duplex paradigms come with a decent security foundation. Indeed, soon after the introduction of sponge functions [BDPV07], Bertoni et al. [BDPV08] proved that, if the permutation p is random, the construction is indifferentiable [MRH04, CDMP05] from a random oracle up to a complexity $2^{c/2}$. This result, consequently [AMP10, Appendix A], means that the sponge construction truncated to an output of n bits achieves collision security up to complexity $\min\{2^{c/2}, 2^{n/2}\}$ and preimage and second preimage security up to complexity $\min\{2^{c/2}, 2^n\}$. In a separate work, Lefevre and Mennink [LM22] recently improved the bound of preimage resistance to $\min\{\max\{2^{c/2}, 2^{n-r}\}, 2^n\}$, therewith assuring that all bounds on keyless hashing of the sponge are tight in the sense that they are matched by attacks already specified in the original specification of sponge functions [BDPV07, Section 5]. These tight results directly apply to the generic security of Ascon-Hash, Ascon-XOF, and Ascon-CXOF (as we also outline in Section 8).

The indifferentiability result implies that one can also use the sponge in keyed applications, but dedicated analyses were performed to obtain more finegrained bounds. The idea of keying the sponge was outlined in detail by Bertoni et al. [BDPV07, BDPV11b], a construction currently known as the outer-keyed sponge. Chang et al. [CDH+12] suggested to key the sponge in the inner part of the initial state, a construction currently known as the inner-keyed sponge. The two constructions were analyzed in depth by Andreeva et al. [ADMV15]. Naito and Yasuda improved the security bounds of these constructions [NY16]. Mennink et al. [MRV15] formalized and analyzed the full-keyed sponge (inspired by DonkeySponge [BDPV12]) that absorbs the plaintext over the full state. An independent analysis, but for fixed-length outputs, was given by Gaži et al. [GPT15]. These analyses together gave a rather complete view of security of the sponge where the state is initialized with the key, with one non-tight step in the outer-keyed sponge, namely when upper bounding an event that captures key prediction, and this gap was closed by Mennink [Men18] in a dedicated analysis. An alternative version of the keyed sponge, namely the version where one keys the state at the end, also appeared in the original specification of Bertoni et al. [BDPV07] but was generalized and analyzed by Dobraunig and Mennink [DM19b, DM20], who in addition proved leakage resilience of the construction. Berendsen and Mennink [BM24] fine-tuned and improved the leakage resilience analysis.

The duplex construction [BDPV11a] received a separate detailed treatment. The security of the original duplex was related to the indifferentiability of the sponge [BDPV11a], and it was used for the description of SpongeWrap authenticated encryption. Mennink et al. [MRV15] considered the full-keyed duplex, whose security was related to that of the full-keyed sponge. Daemen et al. [DMV17] generalized the duplex to a more powerful construction, and Dobraunig and Mennink [DM19a] subsequently proved leakage resilience of the duplex. An excellent systematization of knowledge on the security of the duplex and its implications is given by Mennink [Men23], who also proved security of Ascon-PRF (we will elaborate on this in Section 9), and gave a detailed description of authenticated encryption using a modern duplex variant, called MonkeySpongeWrap.

1.2 Generic Security of Ascon-AE Constructions

Given this state of affairs, it is tempting to state that the security of the Ascon-AE mode immediately follows. Indeed, the Ascon-AE mode resembles SpongeWrap [BDPV11a] or MonkeySpongeWrap [Men23], and those security results give *some* certainty that the Ascon-AE mode is sound. Likewise, along with their security proof of NORX, Jovanovic et al. [JLM14, JLM+19] mentioned that their proof can be generalized to the Ascon-AE mode, though without proof. The main difference between those analyses and the mode of Ascon-AE is the presence of the additional key blindings.

Thus, a dedicated analysis of the Ascon-AE mode, and particular the impact of these

key blindings, turned out to be necessary and relevant, and this has lead to multiple works considering the security of the Ascon-AE mode. Chakraborty et al. [CDN23] performed a single-user security analysis in the nonce-respecting setting [BN00], and independently, Lefevre and Mennink [LM24] delivered a multi-user security analysis in both the nonce-respecting and nonce-misuse setting [RS06]. Soon after their first work, Chakraborty et al. [CDN24] also extended their proof to multi-user security and to the nonce-misuse setting. On top of that, there is a "proof sketch" of Guo et al. [GPPS19b] (full version of [GPPS20]) in the nonce-misuse resilience setting that guarantees security for fresh nonces only [ADL17] and in the leakage resilience setting where the inner permutations may leak side-channel information [DP08, PSV15]. Finally, Lefevre and Mennink [LM24] also included a proof under state recovery, demonstrating that Ascon-AE still achieves authenticity even if the adversary learns all internal states. It should be noted that all these results are in the random permutation model, where the permutation p is assumed to be a random permutation, or (in some of these results) the outer permutation and inner permutation are both random and assumed to be independent (see Remark 1).

1.3 A Decent Classification

From this overall state-of-the-art discussion, it can be concluded that the security analyses of the Ascon-AE mode has lead to many different results, all in different security models, different attack settings, different proof techniques, and in fact also with different levels of accuracy. On top of that, most of these bounds are not matched with tight attacks, which means that we do not know if all the bounds are tight and can be improved. This is a particularly relevant question in the area of lightweight cryptography, where schemes are minimized and a too loose bound would give a false sense of insecurity (as also already mentioned in myriad earlier works [DM20, LNS18, JN20, DDNT23, LMP17, BM24]).

In this work, we give a complete and comprehensive overview of the levels of security of the Ascon-AE mode in various security settings. We cover three flavors of conventional security (in Section 4): nonce-respecting security [BN00], nonce-misuse resistance [RS06], and nonce-misuse resilience [ADL17]. We subsequently cover three flavors of leaky security (in Section 5): bounded leakage resilience in a leveled implementation setup [DP08, PSV15], state-recovery security [LM24], and security under release of unverified plaintext [ABL+14]. For each of these security models, (i) we categorize the existing security lower and upper bounds, (ii) we point out multiple flaws and issues in existing analyses, and (iii) we derive new security bounds and generic attacks to complete the overview. A high-level overview is given in Figure 1.

Most notably, apart from simply classifying existing results, the systematization makes the following contributions:

- We develop new security proofs for nonce-misuse resilience, leakage resilience, and release of unverified plaintext, as these were lacking;
- We also revisit earlier security proofs that were done under the assumption that the outer and inner permutation of Ascon-AE were independent, and adapt them to the single-permutation model;
- We point out a flaw in the nonce-misuse authenticity analysis of Chakraborty et al. [CDN24];
- We give matching attacks for all security proofs.

All new security proofs are gathered together in Section 6, and all elaborate generic attacks in Section 7.

As a bonus, we also comprehensibly discuss how existing literature covers the generic security of Ascon-Hash and Ascon-PRF in Section 8 and Section 9, respectively. These

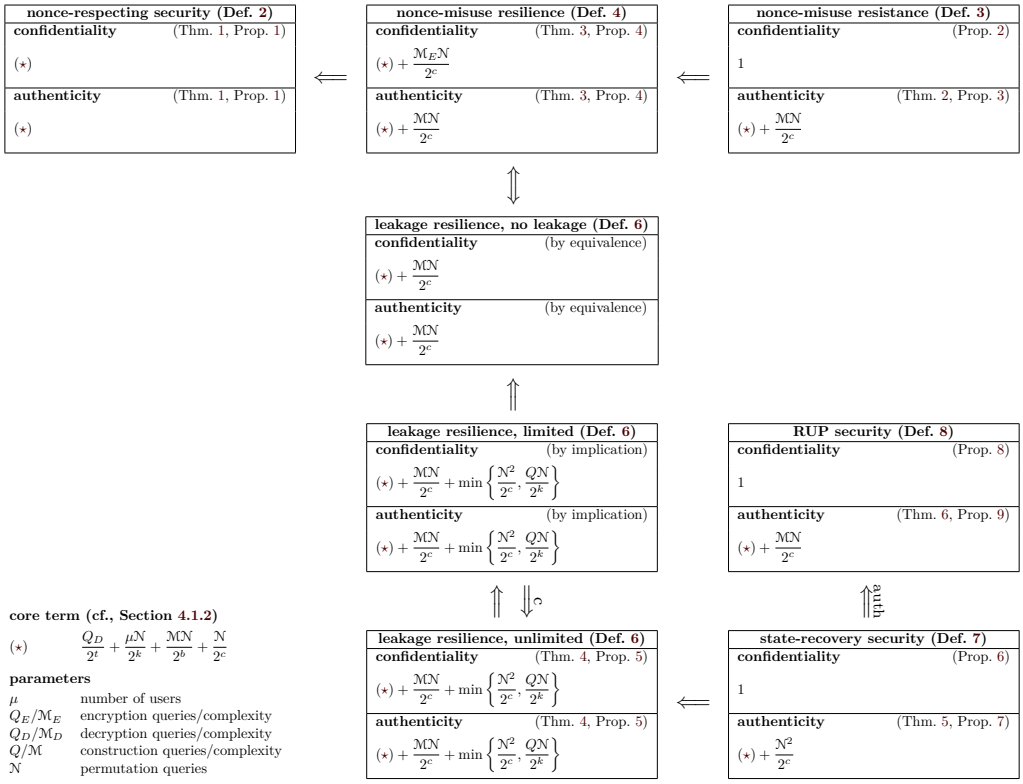

Figure 1: High-level overview of the considered security models and the corresponding results. Intuitively, horizontal orientation represents the amount of nonce-misuse power whereas vertical orientation represents the amount of additional leakage. All bounds are simplified, they are expressed in big $\mathcal{O}$ notation, and are tight in the simplified setting of Section 3.3. The conditional implication $\overset{\text{c}}{\Longrightarrow}$ depends on the set of allowed leakage functions but applies in our case (cf., Section 5.1). The implication $\overset{\text{auth}}{\Longleftarrow}$ only holds for authenticity (cf., Lemma 2).

sections are not so surprising: the generic security of Ascon-Hash follows from the results given in the first paragraph of Section 1.1, and Mennink [Men23] gave a security proof of Ascon-PRF. Finally, we conclude the work in Section 10, where we highlight models or settings that we do not cover and give a further final discussion.

1.4 Outline

We first settle some basic notation in Section 1.5. The Ascon authenticated encryption (Ascon-AE) mode is described in detail in Section 2. We describe the general attack model, including a description of the adversarial resources and some notational conventions in Section 3. Then, in Section 4, we discuss the conventional security models of nonce-respecting security (Section 4.1), nonce-misuse resistance (Section 4.2), and nonce-misuse resilience (Section 4.3). Then, we extend the analysis to security in leaky settings in Section 5, covering leakage resilience (Section 5.1), state-recovery security (5.2), and release of unverified plaintext (5.3). The security proofs are all gathered in Section 6 and the generic attacks in Section 7. We then extend our discussion to Ascon-Hash/Ascon-(C)XOF in Section 8 and to Ascon-PRF in Section 9. We conclude the work in Section 10.

1.5 Notation

Let $a, b \in \mathbb{N}$ such that $a \leq b$. We denote $[\![a, b]\!] = \{a, \ldots, b\}$. We furthermore denote by $\{0,1\}^b$ the set of b-bit strings, by $\{0,1\}^*$ the set of arbitrarily long strings (including the empty string $\emptyset$), by $\left(\{0,1\}^b\right)^*$ the set of bit strings of length a multiple of b (again including the empty string $\emptyset$), and we denote $\{0,1\}^{\leq b} = \bigcup_{i=0}^{b} \{0,1\}^i$. We denote the set of all b-bit permutations $p : \{0,1\}^b \to \{0,1\}^b$ by $\mathrm{Perm}\,(b)$.

We define by pad_s the padding function that gets as input a bit string $X \in \{0,1\}^*$, and that splits it into s-bit blocks, where the last block is of size between 0 and $s-1$ bits. The result is thus a tuple of blocks. We define by $\mathrm{pad}_s^{10^*}(X)$ the padding function that gets as input a bit string $X \in \{0,1\}^*$, that pads it with a 1 and a sufficient number of 0s so that the length becomes a multiple of s bits, and then it splits the resulting string into s-bit blocks. Also here, the result is a tuple of blocks.

For a string or a tuple X, if a, b are such that $1 \leq a < b \leq |X|$, then $X[a:b]$ denotes the substring or subtuple starting at position a and ending at position b. We write $X[a] = X[a:a]$ for brevity. We denote by $\lceil X \rceil_a$ the leftmost a elements and by $\lfloor X \rfloor_a$ the rightmost a elements. For any two strings or tuples X, Y, we denote their concatenation by $X \| Y$ and if $|X| = |Y|$ their bitwise exclusive or (XOR) by $X \oplus Y$. In addition, if $c \leq \min\{|X|, |Y|\}$, $X \overset{c}{=} Y$ means that $\lfloor X \rfloor_c = \lfloor Y \rfloor_c$.

If $\mathcal{S}$ is a set, we denote by $\exists^{\neq} x, y \in \mathcal{S}$ the existence of two distinct elements in $\mathcal{S}$. Moreover, if $\mathcal{S}$ is finite, we denote by $X \xleftarrow{\$} \mathcal{S}$ the uniform random drawing of X from $\mathcal{S}$.

Assuming that $a < b$, the falling factorial of b of depth a is denoted by $(b)_a$.

2 Ascon-AE Mode

The Ascon-AE mode [DEMS21,DEMS14,DEMS19] is a variant of SpongeWrap [BDPV11a], with additional key blinding during the initialization and the finalization phases. Let $b, c, r, k, n, t \in \mathbb{N}$ such that $b = r + c$, $k + n \leq b$, $t \leq k$, $k \leq c$, and let p be a cryptographic permutation over b bits. The Ascon-AE mode is an ensemble of two algorithms, encryption $\mathbf{Enc}^p$ and decryption $\mathbf{Dec}^p$. The encryption algorithm $\mathbf{Enc}^p$ takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^n$, associated data $A \in \{0,1\}^*$, and a plaintext $P \in \{0,1\}^*$. It returns a ciphertext $C \in \{0,1\}^*$ with $|C| = |P|$, and a tag $T \in \{0,1\}^t$. For simplicity of notation, we will put the key as a subscript to $\mathbf{Enc}^p$. Therefore, the encryption algorithm based on the key K and permutation p is denoted as:

$$\begin{aligned} \mathbf{Enc}_K^p : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* &\longrightarrow \{0,1\}^* \times \{0,1\}^t\,, \\ (N, A, P) &\longrightarrow (C \in \{0,1\}^{|P|}, T)\,. \end{aligned}$$

The decryption algorithm $\mathbf{Dec}^p$ takes as input a key $K \in \{0,1\}^k$ (again put as a subscript), a nonce $N \in \{0,1\}^n$, associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^t$. It returns either the corresponding plaintext $P \in \{0,1\}^*$ with $|P| = |C|$ if authentication with the tag succeeds, or a failure symbol $\perp$. Therefore, the decryption algorithm based on the key K and permutation p is denoted as:

$$\begin{aligned} \mathbf{Dec}_K^p : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t &\longrightarrow \{0,1\}^* \cup \{\perp\}\,, \\ (N, A, C, T) &\longrightarrow P \in \{0,1\}^{|C|} \text{ or } \perp\,. \end{aligned}$$

The encryption and decryption algorithms of Ascon-AE are described in Algorithm 1 and illustrated in Figure 2. Here, $IV \in \{0,1\}^{b-k-n}$ is a fixed initialization value encoding the specific instance of Ascon-AE. The Ascon specification [DEMS19] specifies Ascon-AE to always operate with nonce size n and tag size t equal to 128 bits. The basic variant, Ascon-128, has a capacity $c = 256$ and rate $r = 64$, while the accelerated variant, Ascon-128a,

has a capacity $c = 192$ and rate $r = 128$, and both use a key size $k = 128$. The variant Ascon-80pq differs from the basic variant Ascon-128 in the fact that the key is increased to size $k = 160$. The NIST draft standard [SMKK24] specifies a single instance, namely Ascon-AEAD128, which is based on Ascon-128a and shares the same parameters sizes (i.e., n, t, k, r, and c). On top of that, Ascon-AEAD128 supports two implementation options: (i) truncating the tag to a size of up to $t = 64$ bits, and (ii) masking the nonce with an additional 128-bit key.

Remark 1. We would like to remark that the Ascon-AE specification, in fact, operates on two different permutations: an outer permutation p_{o} for the initialization and finalization and an inner permutation p_{i} for the inner evaluations. These permutations differ only in the number of rounds and the round constants, which were carefully chosen by the designers to balance efficiency and security. On the other hand, this work focuses on generic security in the ideal permutation model. Current techniques, however, cannot model the dependency between p_{o} and p_{i}. This leaves us with two options to model p_{o} and p_{i}: either model them as being independent, as Guo et al. [GPPS19b] and Lefevre and Mennink [LM22] did, or model them as being the same, as Chakraborty et al. [CDN23, CDN24] did. Security proofs with independent primitives are often stronger than with identical permutations (we have seen this, for example, for the sum of two secret permutations [BKR98, Luc00, Pat08a, DHT17, Din24]), and this seems to be the case for the Ascon mode, too. For this reason, we chose to model the two permutations as being identical, and consequently, we will also have to redo/update some earlier proofs.

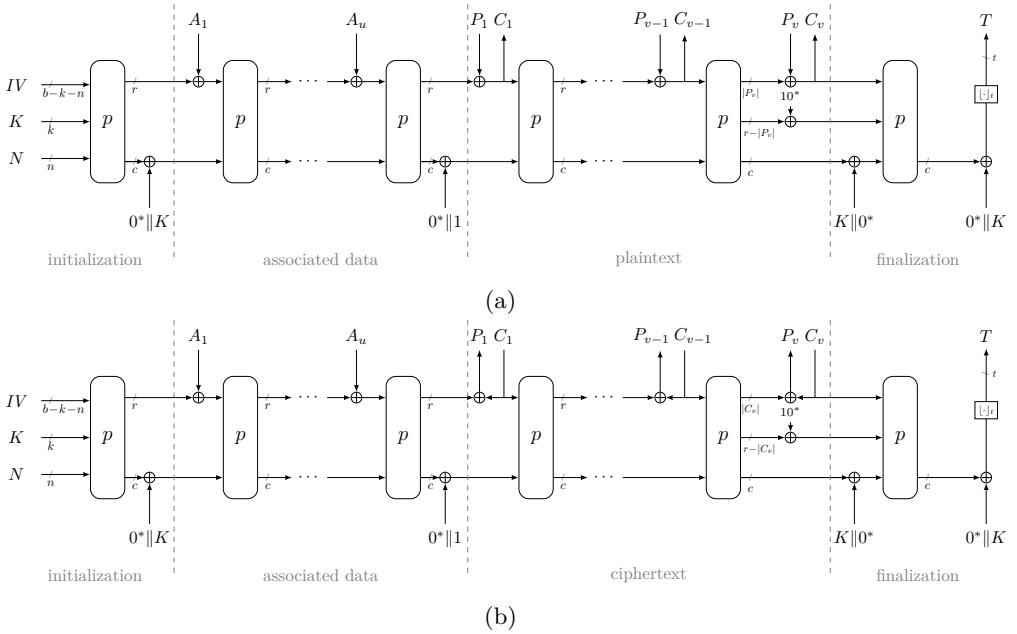

Figure 2: The Ascon-AE mode of operation in case of non-empty associated data: (a) encryption **Enc** and (b) decryption **Dec**. Here, A is injectively padded as $(A_1, \ldots, A_u) \leftarrow \mathrm{pad}_r^{10^*}(A)$. For encryption, the plaintext $P \in \{0,1\}^*$ is padded as $(P_1, \ldots, P_v) \leftarrow \mathrm{pad}_r(P)$, and for decryption the ciphertext $C \in \{0,1\}^*$ is padded as $(C_1, \ldots, C_v) \leftarrow \mathrm{pad}_r(C)$, noting that we put the 10^*-padding explicit in the picture.

Algorithm 1 Ascon-AE mode

Encryption algorithm **Enc**	Decryption algorithm **Dec**
Input:	*Input*:
$K \in \{0,1\}^k$, $N \in \{0,1\}^n$,	$K \in \{0,1\}^k$, $N \in \{0,1\}^n$; $A \in \{0,1\}^*$,
$A \in \{0,1\}^*$, $P \in \{0,1\}^*$	$C \in \{0,1\}^*$, $T \in \{0,1\}^t$
Output:	*Output*:
$C \in \{0,1\}^{\lvert P\rvert}$, $T \in \{0,1\}^t$	Either $P \in \{0,1\}^{\lvert C\rvert}$ or $\perp$
$=\!=\!=$ Initialization $=\!=\!=$	$=\!=\!=$ Initialization $=\!=\!=$
$S \leftarrow p\left(IV \Vert K \Vert N\right) \oplus \left(0^{b-k} \Vert K\right)$	$S \leftarrow p\left(IV \Vert K \Vert N\right) \oplus \left(0^{b-k} \Vert K\right)$
$=\!=\!=$ Absorb A $=\!=\!=$	$=\!=\!=$ Absorb A $=\!=\!=$
if $\lvert A\rvert \geq 1$	**if** $\lvert A\rvert \geq 1$
$(A_1, \ldots, A_u) \leftarrow \mathrm{pad}_r^{10^*}(A)$	$(A_1, \ldots, A_u) \leftarrow \mathrm{pad}_r^{10^*}(A)$
for $i = 1, \ldots, u$ **do**	**for** $i = 1, \ldots, u$ **do**
$S \leftarrow p\left(S \oplus (A_i \Vert 0^c)\right)$	$S \leftarrow p\left(S \oplus (A_i \Vert 0^c)\right)$
$=\!=\!=$ Domain Separation $=\!=\!=$	$=\!=\!=$ Domain Separation $=\!=\!=$
$S \leftarrow S \oplus \left(0^{b-1} \Vert 1\right)$	$S \leftarrow S \oplus \left(0^{b-1} \Vert 1\right)$
$=\!=\!=$ Absorb P, Extract C $=\!=\!=$	$=\!=\!=$ Absorb C, Extract P $=\!=\!=$
$(P_1, \ldots, P_v) \leftarrow \mathrm{pad}_r^{10^*}(P)$	$(C_1, \ldots, C_v) \leftarrow \mathrm{pad}_r^{10^*}(C)$
for $i = 1, \ldots, v-1$ **do**	**for** $i = 1, \ldots, v-1$ **do**
$S \leftarrow S \oplus (P_i \Vert 0^c)$	$P_i \leftarrow \lceil S \rceil_r \oplus C_i$
$C_i \leftarrow \lceil S \rceil_r$	$S \leftarrow C_i \Vert \lfloor S \rfloor_c$
$S \leftarrow p(S)$	$S \leftarrow p\left(S\right)$
$S \leftarrow S \oplus (P_v \Vert 0^c)$	$P_v \leftarrow \lceil \lceil S \rceil_r \oplus C_v \rceil_{\lvert C\rvert \bmod r}$
$C_v \leftarrow \lceil S \rceil_{\lvert P\rvert \bmod r}$	$S \leftarrow S \oplus \left(P_v \Vert 10^{b-1-\lvert P_v\rvert}\right)$
$=\!=\!=$ Finalization $=\!=\!=$	$=\!=\!=$ Finalization $=\!=\!=$
$S \leftarrow p\left(S \oplus \left(0^r \Vert K \Vert 0^{c-k}\right)\right) \oplus \left(0^{b-k} \Vert K\right)$	$S \leftarrow p\left(S \oplus \left(0^r \Vert K \Vert 0^{c-k}\right)\right) \oplus \left(0^{b-k} \Vert K\right)$
$T \leftarrow \lfloor S \rfloor_t$	$T^* \leftarrow \lfloor S \rfloor_t$
return $(C_1 \Vert \cdots \Vert C_v, T)$	**if** $T^* = T$ **return** $P_1 \Vert \cdots \Vert P_v$
	else return $\perp$

3 Adversarial Setup

We will consider the security of Ascon-AE in various attack models, but to describe these models and the levels of security appropriately, we first have to define the concept of adversaries and distinguishing advantages in Section 3.1 and certain conventions in notation in Section 3.2. We then describe how we quantify adversaries in Section 3.3, and present a useful mathematical result on multicollisions in Section 3.4.

3.1 Adversaries and Distinguishing Advantages

An adversary $\mathcal{A}$ is an algorithm. It is given access to a collection of oracles $\mathcal{O}$, which we denote by $\mathcal{A}[\mathcal{O}]$. In our work, we consider two settings.

The first one is where $\mathcal{A}$ operates as distinguisher. In this case, $\mathcal{O}$ is in fact either of two collections of oracles, $\mathcal{W}_0$ or $\mathcal{W}_1$, and $\mathcal{A}$ has to determine which one it communicates with. At the end of its interaction, $\mathcal{A}$ outputs either 0 or 1, and we denote

$$\Delta_{\mathcal{A}}(\mathcal{W}_0\,;\,\mathcal{W}_1) = \left|\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_0\right] \to 1\right) - \mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_1\right] \to 1\right)\right| .$$

The second one is where $\mathcal{A}$ is expected to mount a specific type of attack. In this case, $\mathcal{A}$ knows the set of oracles it communicates with, and succeeds if it fulfills a task that is made explicit in the security definition (in our AE definitions, this will be "mounting a forgery").

The collections of oracles in this work will always be composed of Ascon-AE algorithms, i.e., **Enc** and **Dec** of Section 2 or their random equivalents (and the precise definition of the oracles highly depends on the specific security model). As a matter of fact, we will consider security of Ascon-AE in the multi-user setting, where the adversary has access to μ instances of the construction. Finally, as we will only consider security in the random permutation model, $p \xleftarrow{\$} \mathrm{Perm}(b)$ will always be one of the oracles, to which $\mathcal{A}$ even has forward and inverse access, which we denote by $p^{\pm}$. This means that, typically, $\mathcal{O}$ would be of the following form (in our security games, the number of construction oracles ranges from 1 to 4):

$$\mathcal{O} = \left((\mathcal{O}_{1,m}, \mathcal{O}_{2,m}, \mathcal{O}_{3,m}, \mathcal{O}_{4,m})_{m=1}^{\mu}, p^{\pm}\right) . \tag{1}$$

3.2 Notational Conventions

Given that the oracles to which $\mathcal{A}$ has access are encryption and decryption functionalities of Ascon-AE, or their ideal equivalent, we will need to impose restrictions on nonce-repetition or even query-repetition to avoid trivial attacks. These restrictions depend on the actual security model, but we will define shortcut notation for this, for any typical oracle of the form (1):

- $\mathcal{O}_{i,m} \overset{N}{\not\rightarrow} \mathcal{O}_{j,m}$ means that if $\mathcal{A}$ queries oracle $\mathcal{O}_{i,m}$ with a certain nonce N, then it cannot later query $\mathcal{O}_{j,m}$ with the same nonce N;
- $\mathcal{O}_{i,m} \overset{N}{\not\rightarrow} \mathcal{O}_{j,m}$ means that if $\mathcal{A}$ makes *two* distinct queries to $\mathcal{O}_{i,m}$ with the same nonce N, then it cannot later query $\mathcal{O}_{j,m}$ with the same nonce N;
- $\mathcal{O}_{i,m} \overset{*}{\not\rightarrow} \mathcal{O}_{j,m}$ means that a query to $\mathcal{O}_{i,m}$ cannot later be repeated to $\mathcal{O}_{j,m}$. The definition has slightly different meanings whenever the oracles are encryption or decryption oracles: if $\mathcal{O}_{i,m}$ and $\mathcal{O}_{j,m}$ are both encryption or both decryption oracles, the definition means that $\mathcal{A}$ cannot make the same query to both oracles; if one of them is an encryption and one a decryption oracle, the definition means that $\mathcal{A}$ cannot use the response of oracle $\mathcal{O}_{i,m}$ as input to oracle $\mathcal{O}_{j,m}$.

We remark that we *never* impose anything on nonce-repetition or query-repetition among *different* users, i.e., from user m to m'. We do make one general restriction, though, namely that $\mathcal{A}$ never repeats the exact same query to an oracle. For the primitive access $p^{\pm}$ this additionally means that $\mathcal{A}$ never makes an inverse query for an earlier forward query, or vice versa.

3.3 Adversarial Resources

We always consider an information-theoretic distinguisher $\mathcal{A}$. In terms of a collection of oracles of the form (1), its resources are quantified as follows:

- The total number of queries to $(\mathcal{O}_{1,m}, \mathcal{O}_{2,m}, \mathcal{O}_{3,m}, \mathcal{O}_{4,m})_{m=1}^{\mu}$ is denoted by Q, and the total *online complexity* is denoted by $\mathcal{M}$, which counts the number of "blocks", i.e., the minimal number of permutation evaluations that would be required by the Ascon-AE mode to process the query. The quantities $(Q, \mathcal{M})$ may be refined into encryption complexities $(Q_E, \mathcal{M}_E)$, counting only encryption queries, and decryption complexities $(Q_D, \mathcal{M}_D)$, counting only decryption queries;[1]

[1] Looking ahead, there is one exception to this, namely in security under release of unverified plaintext (Section 5.3), where the decryption functionality is split into an unverified decryption function and a verification function. The notation will be refined there in an ad-hoc way.

- The total number of queries to $p^{\pm}$ is counted by the *offline complexity* $\mathcal{N}$.

Without loss of generality, we assume that $\mu \leq Q_E$, since an oracle that the adversary cannot query is of no use. As a rule of thumb, $\mathcal{M} \ll \mathcal{N}$, as $\mathcal{M}$ is limited by the use case in which Ascon-AE is employed whereas $\mathcal{N}$ is limited by the wealth of the adversary (i.e., its computing power).

When investigating tightness of bounds, we ignore constant factors and logarithmic factors, and furthermore assume that $\mathcal{M}_D \leq \mathcal{M}_E$ and $Q_D \leq Q_E$. The reason behind the latter assumption is that, in real-world protocols [GTW24], connections may be broken once too many failed forgery attempts have been mounted. Of course, in the multi-user setting, this assumption is a bit more debatable, as an adversary may apply a forgery attempt on multiple users, but we only use it in the simplified bounding.

3.4 Multicollisions

Central to the analysis of sponge-based keyed cryptographic functionalities is the concept of multicollisions: security only holds under the condition that it is not too easy for an attacker to obtain a multicollision on the outer r bits of the b-bit state of the sponge. There exist two strategies in current sponge-based literature. One of them was first used in the context of sponges by Jovanovic et al. [JLM14]: they use a simple Stirling approximation to bound the event that a multicollision exceeds a value θ, and subsequently reason on the mode's security under the assumption that the multicollision is at most θ. Omitting details, this results in the following strategy:

$$\mathbf{Pr}\left(\mathbf{success}\right) \leq \mathbf{Pr}\left(\mathbf{success} \mid \mathbf{mult} \leq \theta\right) + \mathbf{Pr}\left(\mathbf{mult} > \theta\right) .$$

In a follow-up work, Jovanovic et al. [JLM+19] improved the multicollision bounds by performing a case distinction depending on the values (r, c), thus improving the second probability of this equation. Daemen et al. [DMV17] introduced the multicollision limit function as a definition specifically tailored towards the sponge/duplex. In a bit more detail, they observed that for a sponge-/duplex-based analysis, the left probability is often of the form $\theta\mathcal{N}/2^c$, so by defining θ such that the right probability is of the form $\theta/2^c$, it can be subsumed within the first term to get a joint term of the form $\theta(\mathcal{N}+1)/2^c$.

An alternative approach is to only compute the *expected* size of the maximum multicollision, $\mathbf{Ex}\left(\mathbf{mult}\right)$, and then bound using the following strategy:

$$\mathbf{Pr}\left(\mathbf{success}\right) = \sum_{\theta} \mathbf{Pr}\left(\mathbf{success} \mid \mathbf{mult} = \theta\right) \mathbf{Pr}\left(\mathbf{mult} = \theta\right) .$$

The expectation on **mult** can then be used, observing that for a sponge-/duplex-based analysis, the left probability is often linear in θ (e.g., of the form $\theta\mathcal{N}/2^c$ as mentioned above). Choi et al. [CLL19] used this approach for the PRF security of truncating a permutation. Their bound is general (it does not consider different parameter settings). Lefevre and Mennink [LM24] slightly improved that bound for their analysis of Ascon-AE. Chakraborty et al. [CDN23] used the same approach for their security analysis of Ascon-AE, but with a fine-tuned bound that distinguishes different parameter setups. Their proof technique basically refines the one of Chakraborty et al. [CJN20]. Depending on the parameter setting, one bound may be better than the other one, but the tightest upper bound applies. We thus present both known bounds on the expected value of **mult** in Lemma 1.

Lemma 1 ([CJN20, CDN23, CLL19, LM24])**.** *Let $q, b, r \in \mathbb{N}$ such that $r < b$ and $R = 2^r$. Suppose we uniformly select a set $\mathcal{S}$ of q distinct elements from $\{0,1\}^b$. Define*

$\mathrm{mucol}_r(\mathcal{S}) = \max_{T\in\{0,1\}^r} |\{S \in \mathcal{S} \,:\, \lceil S\rceil_r = T\}|$, *and let* $\mathrm{mucol}(q,R) = \mathbf{Ex}(\mathrm{mucol}_r(\mathcal{S}))$. *We have*

$$\mathrm{mucol}(q,R) \le \begin{cases} 3 & \text{if } 4 \le q \le \sqrt{R}\,, \\ \frac{4\log_2(q)}{\log_2(\log_2(q))} & \text{if } \sqrt{R} < q \le R\,, \\ 5r\lceil \frac{q}{rR}\rceil & \text{if } R < q\,, \end{cases}$$

$$\text{and } \mathrm{mucol}(q,R) \le \frac{2q}{R} + 3\ln(R) + 4\,.$$

Looking ahead, the bounds that include multicollisions will be presented in an abstract form, i.e., using the term $\mathrm{mucol}(q,R)$. In our simplified discussions of these bounds, we will ignore constant and logarithmic factors and use that $\mathrm{mucol}(q,R) = \mathcal{O}\left(1 + \frac{q}{R}\right)$.

We remark that above two approaches to bound multicollisions are incompatible, but in many cases, one can replace the other (and typically, the expectation approach is tighter). Looking ahead, when we discuss Ascon-PRF in Section 9, we take the bound from Mennink [Men23], which is based on Daemen et al. [DMV17] and thus uses the former method. For consistency between the results in this work, we will in fact update that analysis to work with the expectation approach.

4 Conventional Security of Ascon-AE

We start with the more conventional security notions for authenticated encryption and what level of security the Ascon-AE mode achieves in these models: nonce-respecting security in Section 4.1, nonce-misuse resistance in Section 4.2, and nonce-misuse resilience in Section 4.3.

4.1 Nonce-Respecting Security

4.1.1 Security Model

The most conventional security model for nonce-based authenticated encryption is security in the nonce-respecting setting. The first to formally study this notion were Bellare and Namprempre [BN00, BN08], though in a left-or-right setting where the adversary receives the encryption of either M_0 or M_1. Shrimpton [Shr04] introduced the notion of IND-CCA3 security, which at a high level gives the adversary access to either the encryption and decryption functionality, or to a random oracle that always outputs random responses of expected length and a $\perp$-function that always returns the $\perp$-sign. In this model, the adversary is never allowed to repeat nonces under encryption queries. The notion has reappeared in literature under different terminologies (e.g., [NRS14] called it nAE security). We will simply call it AE security, and adapt it to the multi-user setting in the random permutation model where $p \xleftarrow{\$} \mathrm{Perm}(b)$, similar to [LM24].

Definition 1. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. The AE security of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\,;\, (\$_m, \perp)_{m=1}^{\mu}, p^{\pm}\right),$$

where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{N}{\not\rightarrow} \mathcal{O}_{1,m}$ and $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{2,m}$.

The notion of AE security is related to plain confidentiality and authenticity of the mode, as was also already demonstrated by Bellare and Namprempre [BN00, BN08]. In fact, security proofs for authenticated encryption schemes *can* be derived directly under

the AE security model, as Chakraborty et al. [CDN23, CDN24] did for Ascon-AE, for instance. However, looking ahead, Ascon-AE achieves authenticity but not confidentiality in some of the models discussed in this work. Therefore, we consider confidentiality and authenticity separately whenever possible.

Definition 2. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$.

- The nonce-respecting confidentiality of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon-AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left((\mathbf{Enc}^p_{K_m})_{m=1}^{\mu}, p^{\pm}; (\$_m)_{m=1}^{\mu}, p^{\pm}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{N}{\not\rightarrow} \mathcal{O}_{1,m}$;

- The nonce-respecting authenticity of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[(\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m})_{m=1}^{\mu}, p^{\pm}\right] \text{ forges}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{N}{\not\rightarrow} \mathcal{O}_{1,m}$ and $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{2,m}$. Here, "forges" denotes the event that $\mathcal{A}$ makes a query to one of the oracles $\mathbf{Dec}^p_{K_m}$ that does not return $\perp$.

  Note that authenticity can be equivalently stated as a distance:

$$\Delta_{\mathcal{A}}\left((\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m})_{m=1}^{\mu}, p^{\pm}; (\mathbf{Enc}^p_{K_m}, \perp)_{m=1}^{\mu}, p^{\pm}\right),$$

with the same conditions on nonce- and query-reuse. From this, we can easily conclude that AE security implies confidentiality and authenticity. We repeat the inverse reduction as given by Shrimpton [Shr04]. Let $\mathcal{A}$ be any adversary against the AE security of Ascon-AE. Then, by the triangle inequality,

$$\begin{aligned}
\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}) &= \Delta_{\mathcal{A}}\left((\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m})_{m=1}^{\mu}, p^{\pm}; (\$_m, \perp)_{m=1}^{\mu}, p^{\pm}\right) \\
&\leq \Delta_{\mathcal{A}}\left((\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m})_{m=1}^{\mu}, p^{\pm}; (\mathbf{Enc}^p_{K_m}, \perp)_{m=1}^{\mu}, p^{\pm}\right) \\
&+ \Delta_{\mathcal{A}}\left((\mathbf{Enc}^p_{K_m}, \perp)_{m=1}^{\mu}, p^{\pm}; (\$_m, \perp)_{m=1}^{\mu}, p^{\pm}\right) \\
&\leq \mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon-AE}}(\mathcal{A}') + \mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon-AE}}(\mathcal{A}''),
\end{aligned}$$

for some adversaries $\mathcal{A}'$ and $\mathcal{A}''$ with the same query complexities as $\mathcal{A}$.

4.1.2 Overview

In 2014, Jovanovic et al. [JLM14, JLM+19] analyzed the security of the duplex-based mode NORX [AJN14], providing bounds for both confidentiality and authenticity in the nonce-respecting setting. They mentioned that their analysis can be generalized to Ascon-AE, but they did not provide a proof. Lefevre and Mennink [LM24] made a dedicated analysis, recovered the bounds claimed by Jovanovic et al., i.e., they proved (assuming that $\mu \leq \mathcal{M} \ll \mathcal{N}$),

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathcal{O}\left(\frac{Q_D}{2^t} + \frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{N}^2}{2^b} + \frac{\mathcal{M}_D\mathcal{N}}{2^c}\right).$$

In an independent work, Chakraborty et al. [CDN23] derived a tighter bound in the single user setting, and later [CDN24] extended it to the multi-user setting. We present their result in Theorem 1, leaving the multicollision terms in an abstract form (cf., Section 3.4).

Theorem 1 ([CDN23, CDN24]). *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ (see Section 3.3 for a detailed definition of complexity). We have*

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}) \leq{}& \frac{\mu^2}{2^k} + \frac{2Q_D}{2^t} + \frac{\mathcal{M}_E^2}{2^b} + \frac{\mathcal{M}_D(\mathcal{N} + \mathcal{M}_D)}{2^b} + \frac{\text{mucol}(\mathcal{M}_E, 2^r)(\mathcal{M}_D + \mathcal{N})}{2^c} \\ &+ \frac{\mu(\mathcal{N} + \mathcal{M})}{2^k} + \frac{\text{mucol}(Q_E, 2^t) Q_D}{2^c} + \frac{\text{mucol}(\mathcal{M} + \mathcal{N}, 2^t) Q_D}{2^k} \\ &+ \frac{Q_E^2 + Q_D^2 + Q_E Q_D + (2Q_E + Q_D)(\mathcal{M} + \mathcal{N})}{2^b} + \frac{Q_D(\mathcal{M} + \mathcal{N})}{2^{c+t}} \\ &+ \frac{\text{mucol}\left(Q_E, 2^{b-k}\right)(\mathcal{M} + \mathcal{N})}{2^k}.\end{aligned}$$

Ignoring constant and logarithmic factors, and using that $\text{mucol}(q, R) = \mathcal{O}\left(1 + \frac{q}{R}\right)$ and $\mu \leq \mathcal{M} \ll \mathcal{N}$ (cf., Section 3.3), we obtain

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathcal{O}\left(\frac{Q_D}{2^t} + \frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{M}\mathcal{N}}{2^b} + \frac{\mathcal{N}}{2^c}\right). \tag{$\star$}$$

Throughout this work, we will refer to this term as the *core term*, noting that it contributes to many of the bounds that will follow in the rest of this work. A notable observation is that the term $\mathcal{O}\left(\frac{\mathcal{M}_D\mathcal{N}}{2^c}\right)$ is absent in ($\star$). Indeed, in typical duplex-based authenticated encryption schemes, where keys are only added to the initial state (such as in MonkeySpongeWrap [Men23]), this term appears in the nonce-respecting bounds, along with a tight attack for certain parameter sets: Gilbert et al. [GBKR23] described an attack tailored to precise values of $\mathcal{N}$ and $\mathcal{M}_D$ such that $\mathcal{M}_D\mathcal{N}$ is above 2^c, and Lefevre [Lef24] remarked that taking $\mathcal{N} \approx \mathcal{M}_D \approx 2^{c/2}$ allows their attack to succeed with high probability. For Ascon-AE, and in particular the bound of Theorem 1, this term is absent due to the additional key blindings, that turn out to enhance conventional nonce-respecting security.

The core bound ($\star$) is tight in the simplified setting of Section 3.3, as we show in Proposition 1.

Proposition 1. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exist adversaries $\mathcal{A}_1$ with $Q_D \approx 2^t$, $\mathcal{A}_2$ with $\mathcal{N} \approx 2^k/\mu$, and $\mathcal{A}_3$ with $(\mathcal{M}_E + 2^r)\mathcal{N} \approx 2^b$, such that*

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}_1), \mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}_2), \mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon-AE}}(\mathcal{A}_3) \approx 1.$$

The proof of Proposition 1 is given in Section 7.1.

4.2 Nonce-Misuse Resistance

4.2.1 Security Model

The security model of Section 4.1 restricts the adversary to only use fresh nonces for encryption queries. Rogaway and Shrimpton [RS06] proposed the notion of nonce-misuse resistance, to capture settings where the adversary may have the power to reuse nonces. As, throughout this work, we do not focus on a unified AE security definition but rather on the separated notions of confidentiality and authenticity, we only extend Definition 2 to the nonce-misuse resistance setting. The generalization is straightforward: it mainly consists of dropping the nonce-misuse restrictions.

Definition 3. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \text{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$.

- The nonce-misuse resistance confidentiality of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-m-conf}}_{\text{Ascon-AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathbf{Enc}^{p}_{K_m}\right)_{m=1}^{\mu}, p^{\pm}; (\$_m)_{m=1}^{\mu}, p^{\pm}\right);$$

- The nonce-misuse resistance authenticity of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-m-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[\left(\mathbf{Enc}^{p}_{K_m}, \mathbf{Dec}^{p}_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\right] \text{forges}\right),$$

where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{*}{\not\to} \mathcal{O}_{2,m}$. Here, "forges" denotes the event that $\mathcal{A}$ makes a query to one of the oracles $\mathbf{Dec}^{p}_{K_m}$ that does not return $\perp$.

This is the strongest possible attack setting with respect to nonce-reuse, and it is trivial to observe that nonce-misuse security implies nonce-respecting security. We wish to remark that one-pass schemes, i.e., authenticated encryption functions that make only one pass over the data, are known to impossibly achieve nonce-misuse confidentiality in terms of Definition 3, and this particularly applies to Ascon-AE as we demonstrate in Proposition 2 below.

Proposition 2. *Consider the* Ascon-AE *mode of Section 2 with parameters* b, c, r, k, n, t. *There exists an adversary* $\mathcal{A}$ *with* $Q_E = 2$, *such that*

$$\mathbf{Adv}^{\mu\text{-m-conf}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1.$$

Proof. Let $P_1, P_2 \in \{0,1\}^r$, $N \in \{0,1\}^n$, $m \in [\![1, \mu]\!]$. Consider the following attack:

1. Make an encryption query with user m with input $(N, \emptyset, P_1)$, denote the ciphertext by $C_1 \in \{0,1\}^r$;
2. Make an encryption query with user m with input $(N, \emptyset, P_1 \| P_2)$, denote the ciphertext by $C_1' \| C_2'$ where $C_1', C_2' \in \{0,1\}^r$;
3. If $C_1 = C_1'$ return 0, else 1.

In the real world, a repeated block will always output the same ciphertext block, while in the ideal world, this happens with probability $\frac{1}{2^r}$. This term can be reduced further by repeating the attack or by mounting the attack for longer encryption queries. □

4.2.2 Overview

We now focus on authenticity. Lefevre and Mennink [LM24] derived a nonce-misuse authenticity bound of the order

$$\mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c}\right).$$

Independently, Chakraborty et al. [CDN24] derived a tighter bound:

$$\mathcal{O}\left((\star) + \frac{\mathcal{M}_E^2}{2^c}\right).$$

However, we identify a flaw in Chakraborty et al.'s analysis. In Proposition 3, we show that there exists a forgery attack with a success probability of $\approx \frac{\mathcal{M}_E\mathcal{N}}{2^c}$, which contradicts their bound. The cause of this flaw is that the nonce-misuse resistance analysis of Chakraborty et al. [CDN24] is a fairly direct extension of their nonce-respecting setting analysis [CDN23].

However, in this generalization, some unique aspects of nonce-misuse seem to have been overlooked. In detail, in their proof, the bad event $\mathbf{bad}_5$ is used to compute the probability that there exists a collision between permutation evaluations from encryption queries and permutation evaluations from decryption or permutation queries. In the nonce-misuse setting, the adversary can set the outer part of inner states during encryption queries to a value of its choice, thus this event happens with a probability of form $\mathcal{O}\left(\frac{\mathcal{M}_E\mathcal{N}}{2^c}\right)$, and not $\mathcal{O}\left(\frac{\mathcal{M}_D+\mathcal{N}}{2^c} + \frac{\mathcal{M}_E(\mathcal{M}_D+\mathcal{N})}{2^b}\right)$ as claimed in [CDN24].

Therefore, we consider the bound derived by Lefevre and Mennink. However, their results hold in a model where the outer and inner permutations are assumed to be independent, making it incompatible with the model that we consider (cf., Remark 1). Moreover, two technical subtleties can be noted: (i) their result only holds in the case where the tag size equals to the nonce size, and (ii) when inspecting the bounds and manually separating the terms involving the tag from the ones involving the initial state, an undesirable term of the form $\frac{\mathcal{N}}{2^{b-t}}$ appears. Those two factors make the bound less tight in instances of Ascon-AE that have large or small tag sizes. Addressing point (i) is particularly important, as the NIST standard draft allows tags to be as short as 64 bits [SMKK24].

For those reasons, we derive a bound addressing points (i) and (ii), which can be found in Theorem 2.

Theorem 2 ([LM24], revisited)**.** *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ (see Section 3.3 for a detailed definition of complexity). We have*

$$\mathbf{Adv}^{\mu\text{-m-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\,(\mathcal{M}+\mathcal{N})}{2^k} + \frac{18\mathcal{M}\,(\mathcal{M}+\mathcal{N})}{2^c} + \frac{2Q_D}{2^t}\,.$$

The proof of Theorem 2 is given in Section 6.2.

Using that $\mu \leq \mathcal{M}_E \ll \mathcal{N}$ (cf., Section 3.3), we obtain a bound of the order

$$\mathbf{Adv}^{\mu\text{-m-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c}\right)\,. \tag{2}$$

The bound (2) is tight in the simplified setting of Section 3.3. The attacks from Proposition 1 targeting $(\star)$ also apply here, and below Proposition 3 matches the term $\frac{\mathcal{M}_E\mathcal{N}}{2^c}$.

Proposition 3. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exists an adversary $\mathcal{A}$ with $\mathcal{M}_E\mathcal{N} \approx 2^c$, such that*

$$\mathbf{Adv}^{\mu\text{-m-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1\,.$$

The proof of Proposition 3 is given in Section 7.2. Notably, this generic attack invalidates the bound of Chakraborty et al. [CDN24].

4.3 Nonce-Misuse Resilience

4.3.1 Security Model

In Section 4.2, we already mentioned that nonce-misuse resistance is the strongest possible attack setting with respect to nonce-misuse, and one-pass schemes can never achieve confidentiality in this setting. However, one may argue that such one-pass modes still achieve *some* level of confidentiality, namely confidentiality up to common prefix. This idea was formalized under the notion of online authenticated encryption by Fleischmann et al. [FFL12],

and also used by notable authenticated encryption schemes like COPA [ABL+13]. However, this notion has been heavily debated because nonce-misuse may still be devastating through a trivial attack (and also because of different conceptual reasons) [HRRV15]. Although Hoang et al. [HRRV15] do amend their criticism with an alternative notion for online authenticated encryption security, we will not adopt this notion.

Instead, to define a middle-ground between nonce-respecting security (of Section 4.1) and nonce-misuse resistance (of Section 4.2), we resort to the notion of nonce-misuse resilience of Ashur et al. [ADL17]. At a high level, this notion covers that nonce-misuse only affects encryptions under that nonce, and for new nonces, security is still guaranteed. In a bit more detail, for confidentiality, the adversary gets encryption access in two ways: through *challenge* and *non-challenge* queries, where challenge queries should always be for new nonces and non-challenge queries may be for repeated nonces. For authenticity, the adversary may repeat nonces for encryption, but if a nonce is reused, it may not be used in a forgery attempt anymore.

Definition 4. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}\,(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$.

- The nonce-misuse resilience confidentiality of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-mr-conf}}_{\text{Ascon-AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathbf{Enc}^p_{K_m}, \mathbf{Enc}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\ ;\ \left(\mathbf{Enc}^p_{K_m}, \$_m\right)_{m=1}^{\mu}, p^{\pm}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m}/\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{2,m}$, and $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{1,m}$;

- The nonce-misuse resilience authenticity of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-mr-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[\left(\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\right] \text{ forges}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{N}{\not\to} \mathcal{O}_{2,m}$ and $\mathcal{O}_{1,m} \overset{*}{\not\to} \mathcal{O}_{2,m}$. Here, "forges" denotes the event that $\mathcal{A}$ makes a query to one of the oracles $\mathbf{Dec}^p_{K_m}$ that does not return $\perp$.

It is worth noting that nonce-misuse resilience is indeed situated in-between nonce-respecting security and nonce-misuse resistance, or more technically, it implies nonce-respecting security and it is itself implied by nonce-misuse resistance. This observation will also be used below to argue nonce-misuse resilience of Ascon-AE.

4.3.2 Overview

Guo et al. [GPPS19b] considered nonce-misuse resilience of Ascon-AE under two models, named muCCAmL1 (multi-user Chosen-Ciphertext Attack security with misuse-resilience and Leakage), and muCIML2 (multi-user Ciphertext Integrity with Misuse-resistance and Leakage). The bounds are in a leaky setting, therefore yielding lossy bounds if we are only interested in nonce-misuse resilience. More importantly, as the authors admit, their results are merely proof sketches. Additionally, as pointed out by Lefevre and Mennink [LM24], their proofs contain several incomplete and incorrect steps.

Therefore, we derive our own nonce-misuse resilience security analysis, in Theorem 3.

Theorem 3. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{A}^{\text{conf}}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E)$, and $\mathcal{A}^{\text{auth}}$ with*

complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ *(see Section 3.3 for a detailed definition of complexity). We have*

$$\mathbf{Adv}^{\mu\text{-mr-conf}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{conf}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{M}_E+\mathcal{N}\right)}{2^k} + \frac{18\mathcal{M}_E\left(\mathcal{M}_E+\mathcal{N}\right)}{2^c},$$
$$\mathbf{Adv}^{\mu\text{-mr-auth}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{auth}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{M}+\mathcal{N}\right)}{2^k} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} + \frac{2Q_D}{2^t}.$$

Here, the authenticity bound follows from our nonce-misuse resistance proof of Theorem 2. The confidentiality proof of Theorem 3 is new and is given in Section 6.3.

Using that $\mu \leq \mathcal{M}_E \ll \mathcal{N}$ (cf., Section 3.3), we obtain bounds of the order

$$\mathbf{Adv}^{\mu\text{-mr-conf}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{conf}}\right) = \mathcal{O}\left((\star) + \frac{\mathcal{M}_E\mathcal{N}}{2^c}\right), \tag{3}$$
$$\mathbf{Adv}^{\mu\text{-mr-auth}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{auth}}\right) = \mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c}\right). \tag{4}$$

In particular, nonce-misuse resilience confidentiality and authenticity have a bound of the same order as authenticity in the nonce-misuse resistance setting.

The attacks from Proposition 1 targeting the core term $(\star)$ also apply here, and Proposition 4 matches the term $\frac{\mathcal{M}_E\mathcal{N}}{2^c}$. However, the parametrization of $\mathcal{N}$ and $\mathcal{M}_E$ is not completely free here, as the attack requires $\mathcal{N} \geq 2^{k-t}$. Considering the parameter sets of the instances Ascon-128 and Ascon-128a (see Section 2), the dominating term in the bound is $\frac{\mu\mathcal{N}}{2^k}$ anyway. On the other hand, for the parameter sets of Ascon-80pq, the constraint translates to $\mathcal{N} \geq 2^{32}$, which is more than reasonable. Therefore, in the simplified setting of Section 3.3, the bounds (3) and (4) are tight for meaningful parameter sets.

Proposition 4. *Let* $b, c, r, k, n, t \in \mathbb{N}$ *with* $b = r + c$, $k + n \leq b$, $t \leq k$, *and* $k \leq c$. *Consider the* Ascon-AE *mode of Section 2 with parameters* b, c, r, k, n, t. *There exist two adversaries* $\mathcal{A}^{\text{conf}}$ *and* $\mathcal{A}^{\text{auth}}$ *with* $\mathcal{M}_E\mathcal{N} \approx 2^c$ *and* $\mathcal{N} \geq 2^{k-t}$, *such that*

$$\mathbf{Adv}^{\mu\text{-mr-conf}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{conf}}\right), \mathbf{Adv}^{\mu\text{-mr-auth}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{auth}}\right) \approx 1.$$

A proof of Proposition 4 can be found in Section 7.3.

5 Leakage Security of Ascon-AE

We will consider the security of the Ascon-AE mode in leaky settings, where the adversary may learn some internal information during executions through some implementation mistake or through side-channels. We start with leakage resilience in Section 5.1, followed by state-recovery security in Section 5.2, and release of unverified plaintext in Section 5.3.

5.1 Leakage Resilience

5.1.1 Security Model

Authenticated encryption schemes are implemented on a wide variety of platforms, and particularly lightweight authenticated encryption schemes may be implemented in constrained environments which may leak information. It thus makes sense to analyze the leakage resilience of Ascon-AE. There exist various different models on how to model leakage and how to model oracle access. In this work, we restrict our focus to leakage resilience in the bounded leakage model, as set forth by Dziembowski and Pietrzak [DP08] and adopted in various later works [Pie09, YSPY10, FPS12, SPY+10, DP10]. In this model, the adversary gets access to a leak-free version of the construction which it has to distinguish

from random, exactly as in the models of Section 4, but *in addition* it gets access to a leaky version of the scheme, which it may use to gather information. In this leaky version, we assume a priori that every permutation evaluation $p : X \mapsto Y$ leaks certain information. This leakage is captured through a leakage function $L : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$ for some small value λ. For a function $\mathbf{F}^p$ on top of permutation p, we define $[\mathbf{F}^p]_L$ to be an evaluation of $\mathbf{F}^p$ that additionally leaks $L(X,Y)$ for each evaluation $p : X \mapsto Y$. We assume non-adaptive leakage, where the leakage function is defined prior to the experiment and stays fixed throughout [FPS12]. In detail, we define some set of permitted leakage functions $\mathcal{L} = \{L : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ and expect security for any $L \in \mathcal{L}$.

We finally remark that, in its natural form, Ascon-AE cannot achieve confidentiality and integrity under leakage. The reason is that the leakage function L can be chosen in such a way that, from the first evaluation of p, different portions of the key are leaked for different choices of N (a similar attack is described in a bit more detail in Section 5.1.2). To salvage this, we adopt the notion of leveled implementations as put forward by Pereira et al. [PSV15], and that was also adopted in leakage resilience analyses of authenticated encryption schemes [GPPS19a, BGP+20, GPPS20, GPPS19b]. In the concept of leveled implementations, applied to our context, the outer permutations are strongly protected and do not leak any information, whereas the inner permutations may leak.

In detail, we refine $[\mathbf{F}^p]_L$ to be an evaluation of $\mathbf{F}^p$ that additionally leaks $L(X,Y)$ for each *inner* evaluation $p : X \mapsto Y$. This, finally, leads us to the following model, which is based on Barwell et al. [BMOS17] but with two differences: (i) we translate the model to the random permutation model, and (ii) we do not generalize from conventional nonce-respecting security (Section 4.1) but rather from nonce-misuse resilience (Section 4.3).

Definition 5. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. Let $\mathcal{L}$ be a set of leakage functions. The leakage resilience AE-security of Ascon-AE against an adversary $\mathcal{A}$ with respect to $\mathcal{L}$ is defined as

$$\mathbf{Adv}^{\mu\text{-lr-ae}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}) = \max_{L\in\mathcal{L}} \Delta_{\mathcal{A}}\Bigg(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \mathbf{Dec}^p_{K_m} \right)_{m=1}^{\mu}, p^{\pm}; \\ \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot \right)_{m=1}^{\mu}, p^{\pm} \Bigg),$$

where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{1,m}$, $\mathcal{O}_{1,m}/\mathcal{O}_{2,m}/\mathcal{O}_{3,m} \overset{N}{\not\to} \mathcal{O}_{2,m}$, $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{3,m}$, $\mathcal{O}_{1,m}/\mathcal{O}_{3,m} \overset{N}{\not\to} \mathcal{O}_{4,m}$, and $\mathcal{O}_{2,m} \overset{*}{\not\to} \mathcal{O}_{4,m}$.

Just like in Section 4.1, it is more convenient to consider confidentiality and authenticity separately.

Definition 6. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. Let $\mathcal{L}$ be a set of leakage functions.

- The leakage resilience confidentiality of Ascon-AE against an adversary $\mathcal{A}$ with respect to $\mathcal{L}$ is defined as

$$\mathbf{Adv}^{\mu\text{-lr-conf}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}) = \max_{L\in\mathcal{L}} \Delta_{\mathcal{A}}\Big(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L \right)_{m=1}^{\mu}, p^{\pm}; \\ \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m, \left[\mathbf{Dec}^p_{K_m}\right]_L \right)_{m=1}^{\mu}, p^{\pm} \Big),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{1,m}$, $\mathcal{O}_{1,m}/\mathcal{O}_{2,m}/\mathcal{O}_{3,m} \overset{N}{\not\to} \mathcal{O}_{2,m}$, and $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{3,m}$;

- The leakage resilience authenticity of Ascon-AE against an adversary $\mathcal{A}$ with respect to $\mathcal{L}$ is defined as

$$\mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}) = \max_{L\in\mathcal{L}} \mathbf{Pr}\left(\mathcal{A}\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \mathbf{Dec}^p_{K_m}\right)^{\mu}_{m=1}, p^{\pm}\right] \text{ forges}\right),$$

where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{1,m}$, $\mathcal{O}_{1,m}/\mathcal{O}_{2,m}/\mathcal{O}_{3,m} \overset{N}{\not\to} \mathcal{O}_{2,m}$, $\mathcal{O}_{2,m} \overset{N}{\not\to} \mathcal{O}_{3,m}$, $\mathcal{O}_{1,m}/\mathcal{O}_{3,m} \overset{N}{\not\to} \mathcal{O}_{4,m}$, and $\mathcal{O}_{2,m} \overset{*}{\not\to} \mathcal{O}_{4,m}$.

We can again demonstrate that the notions are equivalent, using a similar reasoning as in Section 4.1 (or, to be precise, using a similar reasoning as Ashur et al. [ADL17, Section 4.4]). In detail, let $\mathcal{A}$ be any adversary against the AE security of Ascon-AE. Then,

$$\begin{aligned}
\mathbf{Adv}^{\mu\text{-lr-ae}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}) &= \max_{L\in\mathcal{L}} \Delta_{\mathcal{A}}\Bigg(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \mathbf{Dec}^p_{K_m}\right)^{\mu}_{m=1}, p^{\pm}; \\
&\qquad \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot\right)^{\mu}_{m=1}, p^{\pm}\Bigg) \\
&\le \max_{L\in\mathcal{L}} \Delta_{\mathcal{A}}\Bigg(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \mathbf{Dec}^p_{K_m}\right)^{\mu}_{m=1}, p^{\pm}; \\
&\qquad \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot\right)^{\mu}_{m=1}, p^{\pm}\Bigg) \\
&\quad + \max_{L\in\mathcal{L}} \Delta_{\mathcal{A}}\Bigg(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot\right)^{\mu}_{m=1}, p^{\pm}; \\
&\qquad \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot\right)^{\mu}_{m=1}, p^{\pm}\Bigg) \\
&\le \mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}') + \mathbf{Adv}^{\mu\text{-lr-conf}}_{\text{Ascon-AE},\mathcal{L}}(\mathcal{A}'') ,
\end{aligned}$$

for some adversaries $\mathcal{A}'$ and $\mathcal{A}''$ with the same query complexities as $\mathcal{A}$.

Note that if we take the model of Definition 6 with no leakage, hence with $\lambda = 0$, the definition is equivalent to nonce-misuse resilience, which was the starting point of our model. Looking ahead, we may then consider the notion for arbitrary limited leakage (for any λ) or the notion for unlimited leakage where $\lambda = b$. Clearly, leakage resilience with no leakage is implied by leakage resilience with limited leakage, which is in turn implied by leakage resilience with unlimited leakage.

Remark 2. In the bounded leakage model,[2] that we adopt, each evaluation of p leaks λ bits non-adaptively. This could be λ bits of the secret state. The intuition behind this modeling is that it upper bounds the total amount of knowledge that an adversary can obtain after repeated evaluations of that permutation. Showing that this assumption is reasonable, is hard, and likely impossible as it is a rather loose bound [DMP22]. A model that does slightly better would be hard-to-invert leakage, which requires that the leakage has the property that, even under knowledge of the leakage, the secret state is hard to guess [DKL09, FH15], but it is a bit harder to work with. A recent approach that got closer to reality was simulatable leakage, where the adversary gets knowledge of either actual leakage or simulated leakage [SPY13], but the instantiation of this approach was demonstrated to be problematic [LMO+14]. Finally, one can opt to not bound leakage

[2]The notion "bounded" is standard terminology and refers to the fact that there *is* a fixed λ. This is subtly different from the terminology "limited" that we adopt in our analysis and that specifies whether the bound λ is smaller or equal to b.

after all, and leave a yet-to-be-determined leakage term in the bound [DMP22]. This term is a function of all knowledge that is gained by the adversary, and actual side-channel analysis is needed to accurately bound this term.[3]

Also for oracle modeling, different approaches exist. Our approach consists of giving the adversary access to a leaky oracle and a leak-free challenge oracle, which it has to distinguish from random. Intuitively, this model captures the idea that, even though the adversary has obtained leakage in the past, *new* evaluations are still secure. This idea somewhat aligns with the idea of nonce-misuse resilience, which is that even though the adversary has misused nonces in the past, evaluations for *new* nonces are still secure. (It is for this reason, that Definition 6 adopts the nonce restrictions from nonce-misuse resilience of Definition 4.) A notable alternative approach is of the work of Guo et al. [GPPS19a], that was also used to argue leakage resilience of TEDT [BGP+20] and the closely related work TETSponge [GPPS20, GPPS19b] (they also claim results on Ascon-AE in this model). This model structurally differs, at a high level, in the fact that also challenge queries leak, but the security games are not described in a real-or-random setting (as this, presumably, would require the disputable notion of simulatable leakage) but rather in the left-or-right setting where the adversary receives the encryption of either M_0 or M_1. In our impression, their model is incomparable to the model that we adopt above, and one model may better capture certain use cases than another model.

5.1.2 Overview

The only analysis of Ascon-AE against leakage is by Guo et al. [GPPS20, GPPS19b]. However, their result is in a different, incomparable, model, as we explained in Remark 2. Also, their proof lacks a certain level of accuracy as pointed out in Section 4.3. We thus set out to derive new leakage resilience evidence for Ascon-AE, in the model of Definition 6. However, in doing to, we remark that, even though the adversary cannot reuse nonces for challenge queries, it can repeatedly use nonces for non-challenge queries. Because of this, *depending on the set of permitted leakage functions* $\mathcal{L}$, Ascon-AE achieves the same level of leakage resilience under limited leakage as under unlimited leakage. Indeed, if leakage is limited to only $\lambda = 1$ bit but any leakage function is allowed, the leakage function can be defined to consider the first $\log_2(b)$ bits and use that bit string as encoding of the bit it will leak, and by making sufficient queries for the same nonce, the whole state gets eventually leaked. We admit that this is a rather liberal and non-realistic leakage function, but will still consider it for the sake of generality. We remark that specifically considering more realistic leakage functions, for example those that leak the Hamming weight of the first byte of the state, significantly adds to the complexity and would be a research problem on its own [BM24, DMMS21].

Because of this, we restrict our focus to leakage resilience under unlimited leakage, and derive nonce-misuse resilience authenticity and confidentiality in Theorem 4.

Theorem 4 (unlimited leakage)**.** *Let* $b, c, r, k, n, t \in \mathbb{N}$ *with* $b = r + c$, $k + n \leq b$, $t \leq k$, *and* $k \leq c$. *Let* $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. *Consider the* Ascon-AE *mode of Section 2 with parameters* b, c, r, k, n, t. *Let* $\mathcal{L}$ *be the set of all leakage functions over* Ascon-AE *that do not leak any information about the two outer permutation calls during the initialization and finalization phases. Let* $\mathcal{A}^{\text{conf}}$ *be an adversary with complexity* $(\mathcal{N}, Q_E, \mathcal{M}_E)$, *and* $\mathcal{A}^{\text{auth}}$ *with complexity* $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ *(see Section 3.3 for a detailed definition of*

[3] Refer to Kalai and Reyzin [KR19] for a discussion on leakage models.

complexity). We have

$$\mathbf{Adv}^{\mu\text{-lr-conf}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{conf}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} + \min\left\{\frac{14Q\left(\mathcal{N}+\mathcal{M}\right)}{2^k}, \frac{4\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M})\cdot \text{mucol}\left(\mathcal{M}+\mathcal{N}, 2^{c-k}\right)}{2^k}\right\},$$

$$\mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{auth}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} + \frac{2Q_D}{2^t} + \min\left\{\frac{14Q\left(\mathcal{N}+\mathcal{M}\right)}{2^k}, \frac{4\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M})\cdot \text{mucol}\left(\mathcal{M}+\mathcal{N}, 2^{c-k}\right)}{2^k}\right\}.$$

The proof of Theorem 4 is given in Section 6.4.

Using that $\mu \leq \mathcal{M}_E \ll \mathcal{N}$ (cf., Section 3.3), we obtain bounds of the order

$$\mathbf{Adv}^{\mu\text{-lr-conf}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{conf}}\right), = \mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c} + \min\left\{\frac{\mathcal{N}^2}{2^c}, \frac{Q\mathcal{N}}{2^k}\right\}\right), \tag{5}$$

$$\mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{auth}}\right) = \mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c} + \min\left\{\frac{\mathcal{N}^2}{2^c}, \frac{Q\mathcal{N}}{2^k}\right\}\right). \tag{6}$$

The attacks from Propositions 1 and 4 targeting respectively the core term $(\star)$ and the term $\frac{\mathcal{M}_E\mathcal{N}}{2^c}$ also apply here, and Proposition 5 below matches the term $\min\left\{\frac{\mathcal{N}^2}{2^c}, \frac{Q_E\mathcal{N}}{2^k}\right\}$, under the constraint $\mathcal{N} \geq 2^{k-t}$, as in Proposition 4. Therefore, in the simplified setting of Section 3.3, the bounds (5) and (6) are tight for meaningful parameter sets.

Proposition 5. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{L}$ be the set of all leakage functions that do not leak any information about the two outer permutation calls. There exist two adversaries $\mathcal{A}^{\text{conf}}$ and $\mathcal{A}^{\text{auth}}$ with $\mathcal{N} = \max\left\{2^{c/2}, 2^k/Q_E, 2^{k-t}\right\}$ such that*

$$\mathbf{Adv}^{\mu\text{-lr-conf}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{conf}}\right), \mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon-AE},\mathcal{L}}\left(\mathcal{A}^{\text{auth}}\right) \approx 1.$$

Here, the adversaries $\mathcal{A}^{\text{conf}}$ and $\mathcal{A}^{\text{auth}}$ make $\min\left\{2^{k-c/2}, Q_E\right\}$ encryption queries.

The proof of Proposition 5 is given in Section 7.4.

5.2 State-Recovery Security

5.2.1 Security Model

The designers of Ascon claimed that, even if the adversary accidentally learns the internal state of an evaluation of Ascon-AE, mounting forgeries or recovering the key is hard. To investigate this notion and derive a proper security bound, Lefevre and Mennink [LM24] formalized the notion of state-recovery authenticity in the context of Ascon-AE. Their notion was inspired by that of permutation-based leakage resilient authenticity (basically, authenticity of Definition 6), but stronger in the sense that they (i) considered unlimited leakage by default and (ii) did not put any restrictions on nonce-misuse. As a consequence of adjustment (ii), the security game does not have to distinguish between leaky and non-leaky oracles, and the non-leaky ones can be dropped. We repeat their notion, below. For completeness, we also include the logical state-recovery confidentiality variant as a direct generalization of Definition 6 (but with the decryption oracle dropped as encryption and decryption leakages give the same information in this case), which admittedly is moot as Ascon-AE is insecure in this model.

Definition 7. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. Let L be the leakage function that leaks all inner permutation calls, excluding the ones during the initialization and finalization phase.

- The state-recovery confidentiality of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-sr-conf}}_{\text{Ascon-AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\ ;\ \left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m\right)_{m=1}^{\mu}, p^{\pm}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{2,m} \overset{*}{\not\rightarrow} \mathcal{O}_{1,m}$, and $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{2,m}$;

- The state-recovery authenticity of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \left[\mathbf{Dec}^p_{K_m}\right]_L\right)_{m=1}^{\mu}, p^{\pm}\right] \text{ forges}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{2,m}$. Here, "forges" denotes the event that $\mathcal{A}$ makes a query to one of the oracles $\mathbf{Dec}^p_{K_m}$ that does not return $\perp$.

To see that state-recovery authenticity implies leakage resilience authenticity with unlimited leakage, observe that any state-recovery adversary $\mathcal{A}'$ can easily simulate the oracles of a leakage resilience adversary $\mathcal{A}$ because $\mathcal{A}'$ is free from any nonce repetition restrictions. A similar observation applies to state-recovery confidentiality compared to leakage resilience confidentiality, with the reminder that state-recovery confidentiality is not achieved for Ascon-AE as we demonstrate in Proposition 6 below.

Proposition 6. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exists an adversary $\mathcal{A}$ with $Q_E = 2$, such that*

$$\mathbf{Adv}^{\mu\text{-sr-conf}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1\,.$$

Proof. As in the nonce-misuse confidentiality setting (i.e., Proposition 2), the adversary is allowed to repeat nonces between the challenge encryption oracle and the non-challenge encryption oracle, and this breaks security. Let $P_1, P_2 \in \{0,1\}^r$, $N \in \{0,1\}^n$, $m \in [\![1, \mu]\!]$. Consider the following attack:

1. Make an encryption query to oracle $\mathcal{O}_{1,m}$ with input $(N, \emptyset, P_1)$. From the leaked state S, one can recover the ciphertext, denoted by $C \in \{0,1\}^r$;
2. Make an encryption query to oracle $\mathcal{O}_{2,m}$ with input $(N, \emptyset, P_1 \| P_2)$, denote the leftmost r bits of the ciphertext by $C' \in \{0,1\}^r$;
3. If $C = C'$ return 0, else 1.

In the real world, C will always be equal to C' while in the ideal world, this happens with probability $\frac{1}{2^r}$. This term can be reduced further by repeating the attack or by mounting the attack for longer encryption queries. □

5.2.2 Overview

The bound derived by Lefevre and Mennink [LM24] is tight, and they presented a matching attack, which we repeat in Proposition 7. However, similar issues as those in the nonce-misuse authenticity setting (cf., Theorem 2) apply here, namely the fact that they assumed the outer and inner permutations to be independent whereas we now assume identical permutations. To address this, we revisit their result by adapting the proof on leakage resilience from Theorem 4. The revisited result is presented in Theorem 5.

Theorem 5 ([LM24], revisited)**.** *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ (see Section 3.3 for a detailed definition of complexity). We have*

$$\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu(\mathcal{N}+\mathcal{M})}{2^k} + \frac{2Q_D}{2^t} + \frac{18\mathcal{M}(\mathcal{M}+\mathcal{N})}{2^c} + \frac{4(\mathcal{M}+\mathcal{N})^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M}) \cdot \text{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k}.$$

The proof of Theorem 5 is given in Section 6.5.

Using that $\mu \leq \mathcal{M}_E \ll \mathcal{N}$ (cf., Section 3.3), we obtain a bound of the order

$$\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon-AE}}\left(\mathcal{A}^{\text{auth}}\right) = \mathcal{O}\left((\star) + \frac{\mathcal{N}^2}{2^c}\right). \tag{7}$$

The attacks from Proposition 1 targeting the core term $(\star)$ also apply here, and Proposition 7 below matches the term $\frac{\mathcal{N}^2}{2^c}$. Therefore, the bound (7) is tight.

Proposition 7 ([LM24])**.** *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exists an adversary $\mathcal{A}$ with $\mathcal{N} \approx 2^{c/2}$ such that*

$$\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1.$$

The proof of Proposition 7 is given in Section 7.5.

5.3 Release of Unverified Plaintext Security

5.3.1 Security Model

Another weakness in typical use cases of authenticated encryption is in applications that (accidentally) release plaintext before the tag is verified. This may happen, for example, in use cases where there is insufficient secure memory to store the message or mistakes/incompletenesses in implementation occur (cf., Efail [PDM+18]). Andreeva et al. [ABL+14] formalized the idea of security under release of unverified plaintext (RUP). In this formalization, the authenticated decryption functionality **Dec** is separated into a pure decryption functionality **D** that outputs the plaintext (without verification) and a verification functionality **V** that verifies the authentication:

$$\begin{aligned} \mathbf{D}^p_K : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t &\longrightarrow \{0,1\}^*, \\ (N, A, C, T) &\longrightarrow P \in \{0,1\}^{|C|}, \\ \mathbf{V}^p_K : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t &\longrightarrow \{\top, \bot\}, \\ (N, A, C, T) &\longrightarrow \top \text{ or } \bot. \end{aligned}$$

RUP confidentiality is covered by *plaintext awareness*, that considers a distinguisher that has access to the encryption functionality, and either the (unverified) decryption or an extractor **Ext** that has knowledge of earlier encryption queries and aims to simulate the **D** functionality.[4] Authenticity is covered by an adversary that gets access to encryption, (unverified) decryption, and the verification function, and wins if it forges.

[4]This is called plaintext awareness 1 (PA1). Andreeva et al. [ABL+14] described the stronger setting of PA2 where **Ext** has no knowledge of earlier encryption queries, but a scheme that is not PA1 secure is also not PA2 secure.

Definition 8. Consider the Ascon-AE mode of Section 2. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$.

- Let $\mathbf{Ext} = (\mathbf{Ext}_m)_{m=1}^{\mu}$ be a family of stateful algorithms. The RUP confidentiality (or, plaintext awareness) of Ascon-AE against an adversary $\mathcal{A}$ with respect to $\mathbf{Ext}$ is defined as

$$\mathbf{Adv}^{\mu\text{-rup-conf}}_{\text{Ascon-AE},\mathbf{Ext}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathbf{Enc}^p_{K_m}, \mathbf{D}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\ ;\ \left(\mathbf{Enc}^p_{K_m}, \mathbf{Ext}_m\right)_{m=1}^{\mu}, p^{\pm}\right),$$

  where $\mathbf{Ext}_m$ has access to the query history made by $\mathcal{A}$ to $\mathcal{O}_{1,m}$;

- The RUP authenticity of Ascon-AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-rup-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[\left(\mathbf{Enc}^p_{K_m}, \mathbf{D}^p_{K_m}, \mathbf{V}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\right] \text{ forges}\right),$$

  where $\mathcal{A}$ is restricted as follows: $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{3,m}$. Here, "forges" denotes the event that $\mathcal{A}$ makes a query to one of the oracles $\mathbf{V}^p_{K_m}$ that does not return $\perp$.

In the resources $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ of Section 3.3, the terms Q_D and $\mathcal{M}_D$ now additionally account for the queries made to the verification oracle $\mathbf{V}$. We remark that there have been follow-up works of Ashur et al. [ADL17] and Chang et al. [CDD+19] who presented RUPAE and AERUP, respectively, with the aim to unify RUP security into one definition. In our work, however, we restrict to considering separate confidentiality and authenticity notions.

We remark that RUP authenticity is implied by state-recovery authenticity (cf., Definition 7). This implication, however, is not immediately clear, so we write it out in detail. (In fact, this applies to any authenticated encryption scheme, but we write it out for Ascon-AE as this is the scope of the work.)

Lemma 2. *Consider the* Ascon-AE *mode of Section 2. Let* $p \xleftarrow{\$} \mathrm{Perm}(b)$ *and* $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. *Let* $\mathcal{A}$ *be a RUP authenticity adversary with complexity* $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$. *There exists a state-recovery authenticity adversary* $\mathcal{A}'$ *with complexity* $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$, *such that*

$$\mathbf{Adv}^{\mu\text{-rup-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \leq \mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon-AE}}(\mathcal{A}').$$

Proof. Consider RUP authenticity adversary $\mathcal{A}$ that gets access to the following oracles:

$$\left(\left(\mathbf{Enc}^p_{K_m}, \mathbf{D}^p_{K_m}, \mathbf{V}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\right).$$

It is restricted to $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{3,m}$. We construct state-recovery adversary $\mathcal{A}'$ that gets access to the following oracles:

$$\left(\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \left[\mathbf{Dec}^p_{K_m}\right]_L\right)_{m=1}^{\mu}, p^{\pm}\right),$$

that is restricted to $\mathcal{O}_{1,m} \overset{*}{\not\rightarrow} \mathcal{O}_{2,m}$, and that will use its oracles to simulate the oracles of $\mathcal{A}$. Note that $\mathcal{A}$ may repeat certain queries whereas $\mathcal{A}'$ may not. To solve this, $\mathcal{A}'$ will maintain a database of its queries to the previous oracles. If $\mathcal{A}'$ is about to repeat a query, it will instead retrieve the result from the database. For the rest, adversary $\mathcal{A}'$ operates as follows:

- **$\mathcal{A}$ makes an oracle query $\mathbf{Enc}^p_{K_m}(N, A, P)$:** Adversary $\mathcal{A}'$ relays the query to $\left[\mathbf{Enc}^p_{K_m}\right]_L$ to obtain (C, T) and all intermediate states. It discards the intermediate states and relays (C, T) to $\mathcal{A}$. It stores (N, A, P, C, T) in a database;

- **$\mathcal{A}$ makes an oracle query $\mathbf{D}^{p}_{K_m}(N, A, C, T)$:**
  - If (N, A, C, T) corresponds to an earlier encryption query (note that $\mathcal{A}$ may relay from its first to second oracle but $\mathcal{A}'$ may not), there must be a unique P such that (N, A, P, C, T) in $\mathcal{A}'$s database. $\mathcal{A}'$ replies with that plaintext P;
  - If (N, A, C, T) does not correspond to an earlier encryption query, $\mathcal{A}'$ queries $\left[\mathbf{Dec}^{p}_{K_m}\right]_L$ on the same inputs, it reconstructs P from the state leakages, and replies with P;
- **$\mathcal{A}$ makes an oracle query $\mathbf{V}^{p}_{K_m}(N, A, C, T)$:** Adversary $\mathcal{A}'$ queries $\left[\mathbf{Dec}^{p}_{K_m}\right]_L$ on the same inputs. If the response is a valid plaintext, $\mathcal{A}'$ replies with $\top$, otherwise it replies with $\bot$;
- **$\mathcal{A}$ makes an oracle query $p^{\pm}$:** Adversary $\mathcal{A}'$ simply relays the query to its own permutation oracle $p^{\pm}$ and relays the response back.

If $\mathcal{A}$ mounts a forgery in any of its queries to $\mathbf{V}^{p}_{K_m}$, then $\mathcal{A}'$ also mounts a valid forgery. This proves the claim. □

It does not seem possible to reduce RUP confidentiality to state-recovery confidentiality. There is some similarity, though: RUP confidentiality gives the adversary access to an encryption oracle and a decryption oracle, whereas the state-recovery adversary gets access to a leaky encryption oracle (which it can use to simulate the RUP encryption oracle) and a challenge encryption oracle (which it can use to simulate the RUP decryption oracle, noting that it can reuse nonces). However, the RUP adversary is allowed to freely relay queries whereas the state-recovery adversary is not, and in fact upcoming RUP confidentiality attack of Proposition 8 exploits this, and it cannot be turned into a state-recovery confidentiality attack.

5.3.2 Overview

In their formalism of release of unverified plaintext, Andreeva et al. [ABL+14] also demonstrated that nonce-based length-preserving (i.e., $|C| = |P|$) authenticated encryption schemes cannot achieve PA1 security. We repeat their result in the context of Ascon-AE.

Proposition 8. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exists an adversary $\mathcal{A}$ with $Q_E = 1$ and $Q_D = 1$, such that*

$$\mathbf{Adv}^{\mu\text{-rup-conf}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1\,.$$

Proof. Let $C \in \{0,1\}^r$, $N \in \{0,1\}^n$, and $m \in [\![1, \mu]\!]$. Consider the following attack:

1. Make a decryption query with user m with input $(N, \emptyset, C)$, denote the plaintext by $P \in \{0,1\}^r$;
2. Make an encryption query with user m with input $(N, \emptyset, P)$, denote the ciphertext by $C' \in \{0,1\}^r$;
3. If $C = C'$ return 0, else 1.

In the real world, these are identical evaluations of Ascon-AE and $C = C'$, whereas in the ideal world, this only holds if $\mathbf{Ext}_m$ output the right plaintext P and this happens with probability $\frac{1}{2^r}$. This term can be reduced further by repeating the attack or by mounting the attack for a longer decryption query. □

That said, Ascon-AE achieves authenticity under release of unverified plaintext. This already follows from Theorem 5 and Lemma 2, but this bound is not tight and we derive a better bound below.

Theorem 6. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Let $\mu, \mathcal{N}, \mathcal{M}_E, \mathcal{M}_D, Q_E, Q_D \in \mathbb{N}$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$ (see Section 3.3 for a detailed definition of complexity). We have*

$$\mathbf{Adv}^{\mu\text{-rup-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\,(\mathcal{M}+\mathcal{N})}{2^k} + \frac{18\mathcal{M}\,(\mathcal{M}+\mathcal{N})}{2^c} + \frac{2Q_D}{2^t}\,.$$

The proof of Theorem 6 is given in Section 6.6.

Using that $\mu \leq \mathcal{M}_E \ll \mathcal{N}$ (cf., Section 3.3), we obtain a bound of the order

$$\mathbf{Adv}^{\mu\text{-rup-auth}}_{\text{Ascon-AE}}(\mathcal{A}) = \mathcal{O}\left((\star) + \frac{\mathcal{M}\mathcal{N}}{2^c}\right)\,. \tag{8}$$

In particular, RUP authenticity has a bound of the same order as that of nonce-misuse resistance authenticity.

The adversary is more powerful in the RUP setting than in the nonce-misuse resistance setting: the attack described in Proposition 3 applies, where encryption queries can be substituted for decryption queries. Therefore, the bound (8) is tight, as we make explicit in Proposition 9.

Proposition 9. *Let $b, c, r, k, n, t \in \mathbb{N}$ with $b = r + c$, $k + n \leq b$, $t \leq k$, and $k \leq c$. Consider the* Ascon-AE *mode of Section 2 with parameters b, c, r, k, n, t. There exists an adversary $\mathcal{A}$ with $\mathcal{M}\mathcal{N} \approx 2^c$, such that*

$$\mathbf{Adv}^{\mu\text{-rup-auth}}_{\text{Ascon-AE}}(\mathcal{A}) \approx 1\,.$$

6 Security Proofs for Ascon-AE

We include the security proofs of Theorems 2–6 here, in Sections 6.2–6.6, respectively. The first proof, that of Section 6.2, is worked out in full detail as it lays the foundation for the subsequent proofs. Before doing so, we introduce the H-coefficient technique in Section 6.1.

6.1 H-Coefficient Technique

We present below the H-Coefficient technique by Patarin [Pat91, Pat08b], as modernized by Chen and Steinberger [CS14]. Consider two collections of oracles $\mathcal{W}_I$ and $\mathcal{W}_R$, and an adversary $\mathcal{A}$ that aims to distinguish between $\mathcal{W}_I$ and $\mathcal{W}_R$. We summarize the interaction between $\mathcal{A}$ and the world in a transcript, which contains tuples of query-responses. Consider a partition of all the transcripts attainable in $\mathcal{W}_I$ as $\mathcal{T} = \mathcal{T}_{\text{GOOD}} \cup \mathcal{T}_{\text{BAD}}$. If there exist $\epsilon_1, \epsilon_2 \geq 0$ such that

$$\begin{aligned} \forall \tau \in \mathcal{T}_{\text{GOOD}},\ \frac{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right]\ \text{generates}\ \tau\right)}{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right]\ \text{generates}\ \tau\right)} &\geq 1 - \epsilon_1\,, \\ \text{and}\quad \mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right]\ \text{generates}\ \tau \in \mathcal{T}_{\text{BAD}}\right) &\leq \epsilon_2\,, \end{aligned}$$

then,

$$\Delta_{\mathcal{A}}(\mathcal{W}_I, \mathcal{W}_R) \leq \epsilon_1 + \epsilon_2\,.$$

We will use the H-coefficient technique in all subsequent proofs.

6.2 Proof of Theorem 2

Let $\mathcal{A}$ be a nonce-misuse adversary that makes at most $\mathcal{N}$ permutation queries, Q_E encryption queries of at most $\mathcal{M}_E$ blocks, and Q_D decryption queries of at most $\mathcal{M}_D$ blocks, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}^{\mu\text{-m-auth}}_{\text{Ascon}}(\mathcal{A})$. We will adopt a distinguishing game approach, i.e., consider the distinguishing game version of authenticity, where the challenge decryption oracle is replaced by $\perp$. Therefore, the adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\mathbf{Enc}^p_{K_m}, \mathbf{Dec}^p_{K_m}\right)^{\mu}_{m=1}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\mathbf{Enc}^p_{K_m}, \perp\right)^{\mu}_{m=1}, p^{\pm}\right]$. Without loss of generality, we assume in all subsequent proofs that the associated data and plaintext provided as input to the construction oracles are already padded, and if not, the construction oracles return $\perp$.

Transcript Notation. We define notation for the transcript that can be obtained from the adversarial interaction with the different oracles. The transcript, named τ, is an ordered list of tuples. Each tuple registers a query made to an oracle, and its structure depends on the type of query:

- A forward (resp., inverse) permutation query with input X and output Y generates the transcript element (X, Y, fwd) (resp., (X, Y, inv));
- An encryption query with user m, input (N, A, P), and output (C, T) generates the transcript element (E, m, N, A, P, C, T);
- A decryption query with user m, input (N, A, C, T), and output $\tilde{P}$ generates the transcript element $(D, m, N, A, C, T, \tilde{P})$.

Paths Notation. In both worlds, the encryption queries generate intermediate states through permutation evaluations; in the real world, decryption queries also produce intermediate states. In order to label these states properly, let us define some notation. Any element $(O, m, N, A, B, -, -) \in \tau$ is associated to a *path*, denoted by $\texttt{path} = (O, m, N, A, B, 1)$. Here B is equal to the plaintext blocks if $O = E$, or the ciphertext blocks if $O = D$. The last element is a bit; it equals 1 if the path is *final*, so that the final key blinding has been applied to the last permutation call. From $\texttt{path}$, we define inductively subpaths. We say that $\texttt{path}' = (O', m', N', A', B', f')$ is a *parent* of $\texttt{path} = (O, m, N, A, B, f)$ whenever $(O', m', N') = (O, m, N)$, $f' = 0$, and

- either $A' = A$, $|B'| = |B| - 1$, and $B' = B[1 : |B'|]$,
- or $B' = B = \emptyset$, $|A'| = |A| - 1$, and $A' = A[1 : |A'|]$.

The only paths without parents are of the form $(O, m, N, \emptyset, \emptyset, 0)$, while all other paths have a unique parent. Note that either $\texttt{path}$ and $\texttt{path}'$ are both encryption paths (i.e., their first element is E), or both decryption paths (i.e., their first element is D). For convenience, given a construction query, the set defined by the path $\texttt{path}$ associated with this query, along with all ancestors of $\texttt{path}$, is referred to as the set of paths generated by that query. Denote by $\mathcal{P}$ the set of all paths which are generated by all queries. According to the inductive definition above, the number of encryption paths is equal to $\mathcal{M}_E$, the number of decryption paths is equal to $\mathcal{M}_D$, and $|\mathcal{P}| = \mathcal{M}_E + \mathcal{M}_D$. To illustrate this, consider the example encryption query made in Figure 3, with input $(m, N, (A_1, \ldots, A_u), (P_1, \ldots, P_v))$. The final path associated to this encryption query is called $\texttt{path}_4$, which is a child of $\texttt{path}_3$, which is itself a grand$^{\text{X}}$-child of $\texttt{path}_2$ for some $X \geq 0$, which is itself a grand$^{\text{X'}}$-child of $\texttt{path}_1$ for some $X' \geq 0$, which has itself no parent.

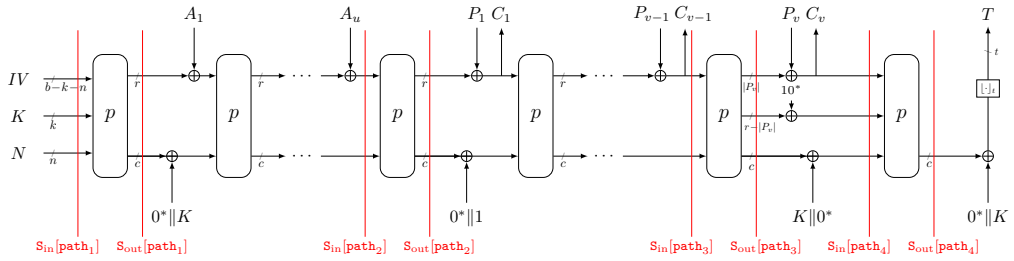

Figure 3: Illustration of intermediate states in the real world of an encryption query. Assuming that the key is the one of user number m, in our example we have $\mathtt{path}_1 = (E, m, N, \emptyset, \emptyset, 0)$, $\mathtt{path}_2 = (E, m, N, (A_1, \ldots, A_u), \emptyset, 0)$, $\mathtt{path}_3 = (E, m, N, (A_1, \ldots, A_u), (P_1, \ldots, P_{v-1}), 0)$, and $\mathtt{path}_4 = (E, m, N, (A_1, \ldots, A_u), (P_1, \ldots, P_v), 1)$.

Intermediate States. We can now effectively label the intermediate states generated by permutation evaluations from encryption queries, and in the real world additionally from decryption queries. We define two dictionaries $\mathtt{S}_{\mathrm{in}}$ and $\mathtt{S}_{\mathrm{out}}$, with labels either in $\mathcal{P}$ in the real world, or in the set of encryption paths in the ideal world. Given $\mathtt{path} = (O, m, N, A, B, f)$, $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ (resp., $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$) corresponds to the input (resp., output) of the permutation evaluation made when doing a construction query with user m, nonce N, after having absorbed the associated data blocks A, and having processed the blocks B (to clarify, if $\mathtt{path}$ is an encryption path, those blocks are considered as being added to the outer part of the state, while if $\mathtt{path}$ is a decryption path, the blocks overwrite the outer parts of the state). In particular, if $\mathtt{path} = (O, m, N, A, P, 1)$, then $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ must include the key addition, and if $\mathtt{path} = (O, m, N, \emptyset, \emptyset, 0)$, then $\mathtt{S}_{\mathrm{in}}[\mathtt{path}] = IV\|K_m\|N$. Figure 3 illustrates $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ and $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$ given our example encryption query.

We will specify a procedure to generate mock intermediate states in the ideal world, but before that we need to introduce further notation to pinpoint the existing relationships between two intermediate states in the real world. First, we say that a decryption path $\mathtt{path} = (D, m, N, A, C, f)$ is *superseded* by an encryption path $\mathtt{path}' = (E, m, N, A, P, f)$ associated to an encryption query $(E, m, N, A', P', C', T) \in \tau$ if $|P| = |C|$ and $C = C'[1 : |C|]$. Intuitively, this means that $\mathtt{path}$ gets the exact same user, nonce, and ciphertexts as an encryption path $\mathtt{path}'$, so that their intermediate states $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ and $\mathtt{S}_{\mathrm{in}}[\mathtt{path}']$ (resp., $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$ and $\mathtt{S}_{\mathrm{out}}[\mathtt{path}']$) must be the same. Let $\mathcal{P}_{\mathrm{S}}$ be the set of decryption paths that are superseded.

Moreover, given two paths $\mathtt{pathP}$ and $\mathtt{pathC}$, $\mathtt{S}_{\mathrm{out}}[\mathtt{pathP}]$ and $\mathtt{S}_{\mathrm{in}}[\mathtt{pathC}]$ must be related whenever $\mathtt{pathP}$ is a parent of $\mathtt{pathC}$, through the addition of constants, key material, and data blocks. We define the function $\mathrm{XorState}\,(\mathtt{pathP}, \mathtt{pathC}) \in \{0,1\}^b$ for any parent-child pair $(\mathtt{pathP}, \mathtt{pathC}) \in \mathcal{P}^2$, which handles the inner parts as follows:

$$\mathrm{XorState}\,((O, m, N, A, B, f), (O, m, N, A', B', f')) =$$
$$\left\{\begin{array}{ll} 0^{b-k}\|K_m & \text{if } A = B = \emptyset \\ 0^b & \text{otherwise} \end{array}\right\} \oplus \left\{\begin{array}{ll} 0^{b-1}1 & \text{if } A = A', B = \emptyset \\ 0^b & \text{otherwise} \end{array}\right\} \oplus \left\{\begin{array}{ll} 0^r\|K_m\|0^{c-k} & \text{if } f' = 1 \\ 0^b & \text{otherwise} \end{array}\right\} .$$

Therefore, we have

$$\mathtt{S}_{\mathrm{in}}[\mathtt{pathC}] \stackrel{c}{=} \mathtt{S}_{\mathrm{out}}[\mathtt{pathP}] \oplus \mathrm{XorState}\,(\mathtt{pathP}, \mathtt{pathC}) \ .$$

Before moving on, we need one last piece of notation. Let $\mathtt{path} = (O, m, N, A, P, f) \in \mathcal{P}$. We define the set $\mathrm{ValidXor}\,(\mathtt{path})$ of b-bit elements. This set captures all the possible values for the inner part of $\mathtt{S}_{\mathrm{in}}[\mathtt{path}']$, for all potential child / superseded paths $\mathtt{path}'$ of

$\mathtt{path}$. It is defined as:

$$\text{ValidXor}\,(\mathtt{path}) = \begin{cases} \{0^b\} & \text{if } f = 1\,, \\ \big\{0^*\|K_m, (0^*\|K_m) \oplus (0^*\|1)\,, & \\ \quad (0^*\|K_m) \oplus (0^*\|1) \oplus \big(0^r\|K_m\|0^{c-k}\big)\big\} & \text{if } A = P = \emptyset\,, \\ \big\{0^b, 0^*\|1, (0^*\|1) \oplus \big(0^r\|K_m\|0^{c-k}\big)\big\} & \text{if } A \neq \emptyset, P = \emptyset\,, \\ \big\{0^b, \big(0^r\|K_m\|0^{c-k}\big)\big\} & \text{if } P \neq \emptyset\,. \end{cases}$$

Looking ahead, ValidXor $(\mathtt{path})$ will be useful for the probability computation of the bad events in a query-wise fashion. An important remark for this upcoming bad event analysis is that $|\text{ValidXor}\,(\mathtt{path})| \leq 3$ always holds.

Mock Intermediate States. Our approach will be to establish an extended transcript that releases the intermediate states associated to all construction queries. Therefore, we define a procedure to generate mock intermediate states in the ideal world, in other words define $\mathtt{S}_{\text{in}}[\mathtt{path}]$ and $\mathtt{S}_{\text{out}}[\mathtt{path}]$ for any decryption path $\mathtt{path} \in \mathcal{P}$. Those states mimic the structure of Ascon-AE by using the intermediate states generated by the prior encryption queries, the user keys $(K_m)_{m=1}^{\mu}$, as well as some fresh randomness. The sampling procedure, taking place at the end of the interaction, operates as follows:

- Sample $\mathtt{S}_{\text{out}}[\mathtt{path}]$, for all $\mathtt{path} = (D, m, N, A, C, f)$ as follows:
  - If $\mathtt{path}$ is superseded by an encryption path $\mathtt{path}'$, then $\mathtt{S}_{\text{out}}[\mathtt{path}] \leftarrow \mathtt{S}_{\text{out}}[\mathtt{path}']$;
  - Else, $\mathtt{S}_{\text{out}}[\mathtt{path}] \xleftarrow{\$} \{0,1\}^b$;
- Then, sample $\mathtt{S}_{\text{in}}[\mathtt{path}]$, for all $\mathtt{path} = (D, m, N, A, C, f)$ as follows:
  - If $P = A = \emptyset$, then necessarily $f = 0$ and $\mathtt{S}_{\text{in}}[(D, m, N, \emptyset, \emptyset, 0)] \leftarrow IV\|K_m\|N$;
  - Else, $\mathtt{path}$ has necessarily a decryption parent path $\mathtt{path}'$. Then:
    - If $C \neq \emptyset$, let $\tilde{C}$ be the last block of C. Then:
    $$\mathtt{S}_{\text{in}}[\mathtt{path}] \leftarrow \tilde{C}\|\lfloor \text{XorState}\,(\mathtt{path}', \mathtt{path}) \oplus \mathtt{S}_{\text{out}}[\mathtt{path}']\rfloor_c\,;$$
    - Else, let $\tilde{A}$ be the last block of A. Then:
    $$\mathtt{S}_{\text{in}}[\mathtt{path}] \leftarrow \big(\tilde{A}\|0^c\big) \oplus \text{XorState}\,(\mathtt{path}', \mathtt{path}) \oplus \mathtt{S}_{\text{out}}[\mathtt{path}']\,.$$

All these states are generated *after* the interactive phase. However, some of them might use randomness from the non-interactive phase (e.g., the $\mathtt{S}_{\text{in}}[(D, m, N, \emptyset, \emptyset, 0)]$s are fixed by the user key K_m). This sampling procedure generates $\mathcal{M}_D - |\mathcal{P}_{\text{S}}|$ b-bit random states.

Extended Transcript. Now that the dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$ are defined for all labels in $\mathcal{P}$ in both worlds, we can define the extended transcript $\tilde{\tau}$ built from τ by adding elements as follows:

- All permutation queries (X, Y, d) are kept untouched;
- A construction query tuple $(O, m, N, B, -) \in \tau$ is followed by tuples of the form $(\mathtt{path}, \mathtt{S}_{\text{in}}[\mathtt{path}], \mathtt{S}_{\text{out}}[\mathtt{path}])$ for all $\mathtt{path}$s generated by the aforementioned query. If a path repeats due to the nonce-misuse setting, the repeating tuples are removed from the transcript. Moreover, any decryption path that is superseded by an encryption path is removed from the transcript;
- At the end of $\tilde{\tau}$, a tuple containing the keys $(K_1, \ldots, K_\mu)$ is added.

Summarizing, from $\tilde{\tau}$ we can reconstruct the list of all permutation queries (X, Y, d), the user's keys, the two dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$, and the sets $\mathcal{P}$ and $\mathcal{P}_{\text{S}}$. This extended transcript is released at the end of the interaction, right before the distinguisher outputs its decision bit.

Bad Events. We introduce the following bad events:

$$
\begin{aligned}
\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}} &: \exists^{\neq} m_1, m_2 \in [\![1,\mu]\!] \text{ such that } K_{m_1} = K_{m_2}\,;\\
\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}} &: \exists m \in [\![1,\mu]\!], (X,Y,\mathrm{d}) \in \tilde{\tau} \text{ such that } X[b-k-n+1:b-n] = K_m, \text{ or}\\
&\quad \exists m \in [\![1,\mu]\!], \mathtt{path}' = (m', N', A', P', f') \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{S}}\\
&\quad \text{such that } (A' \| P' \neq \emptyset) \text{ and } \mathtt{S}_{\mathrm{in}}[(\mathtt{path}')][b-k-n+1:b-n] = K_m\,;\\
\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}} &: \exists^{\neq} \mathtt{path} = (m, N, A, P, 0), \mathtt{path}' = (m', N', A', P', 0) \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{S}}\,,\\
&\quad \exists \delta \in \mathrm{ValidXor}\,(\mathtt{path})\,, \delta' \in \mathrm{ValidXor}\,(\mathtt{path}')\\
&\quad \text{such that } \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \oplus \delta \stackrel{c}{=} \mathtt{S}_{\mathrm{out}}[\mathtt{path}'] \oplus \delta'\,;\\
\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}} &: \exists (X,Y,\mathrm{d}) \in \tilde{\tau}, \mathtt{path} = (O, m, N, A, P, f) \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{S}}, \delta \in \mathrm{ValidXor}\,(\mathtt{path})\\
&\quad \text{such that } \left(f = 0 \text{ and } X \stackrel{c}{=} \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \oplus \delta\right) \text{ or } Y = \mathtt{S}_{\mathrm{out}}[\mathtt{path}]\,;\\
\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}} &: \exists (D, m, N, A, C, T, \tilde{P}) \in \tilde{\tau} \text{ such that } \lfloor \mathtt{S}_{\mathrm{out}}[(D, m, N, A, C, 1)] \rfloor_t \oplus \lfloor K_m \rfloor_t = T, \text{ or}\\
&\quad \exists \mathtt{path} \in \mathcal{P} \text{ superseding } (D, m, N, A, C, 1) \text{ with } \lfloor \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \rfloor_t \oplus \lfloor K_m \rfloor_t = T\,;\\
\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}} &: \mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}} \vee \mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}} \vee \mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}} \vee \mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}} \vee \mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}\,.
\end{aligned}
$$

The sub-/superscript in the bad events indicates the proof setting, where "$\mathbf{m}$" indicates that we are in the nonce-misuse resistance setting and "$\mathbf{a}$" that we focus on authenticity. (Here, we remark that upcoming proofs extend over this main proof, and so do their bad events.) $\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}}$ pinpoints collisions between two user keys, $\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}$ corresponds to the event that the adversary guesses a user key via a permutation query, or indirectly via a construction query. $\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}$ refers to the event that two potential intermediate states collide on their inner parts, and $\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}}$ corresponds to the adversary guessing an intermediate (or potential future intermediate) state. Finally, $\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}$ corresponds to the event that a decryption query is rejected, but the corresponding intermediate state returns the tag that was guessed by the adversary. This covers two situations, as outlined in the bad event: (i) a decryption query with a tag T is made, but the corresponding mock final state generated at the end of the interaction matches the tag, and (ii) a decryption query with a tag T is made, but later the corresponding encryption is made and returns the tag T.

Probability of Good Transcripts. As long as $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ is not set, there are no collisions between intermediate states, and no overlap exists between the permutation evaluations stemming from construction queries and those from permutation queries. Consequently, by the design of the sampling procedure in the ideal world, the intermediate states generated in the ideal world adhere to the structure of the mode Ascon-AE. Moreover, $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ guarantees that the real world rejects all decryption queries. Therefore, every good transcript which is reachable in the real world is also reachable in the ideal world, and vice-versa.

Let $\tilde{\tau}$ be a transcript that does not set $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$. In the real world, this transcript might not induce exactly $\mathcal{N} + \mathcal{M}_E + \mathcal{M}_D$ permutation calls, as the permutation evaluations from encryption and decryption queries might overlap.[5] Let $\ell_E(\tilde{\tau})$, $\ell_D(\tilde{\tau})$ be such that, in the real world, the encryption (resp., decryption) queries induce exactly $\mathcal{M}_E - \ell_E(\tilde{\tau})$ (resp., $\mathcal{M}_D - \ell_D(\tilde{\tau})$) distinct permutation evaluations, and let $\ell(\tilde{\tau}) = \ell_E(\tilde{\tau}) + \ell_D(\tilde{\tau})$. We have

$$
\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right) = \frac{1}{(2^b)_{\mathcal{N}+\mathcal{M}_E+\mathcal{M}_D-\ell(\tilde{\tau})}} \frac{1}{(2^k)^{\mu}}\,.
$$

[5] Indeed, the quantities $\mathcal{M}_E$ and $\mathcal{M}_D$ are defined separately, but do not account for potential repeated permutation evaluations. For instance, if encryption query (E, m, N, A, P, C, T) is followed by a decryption query $(D, m, N, A, C\|\tilde{C}, \tilde{T})$, then fresh permutation evaluations begin only from the moment of absorbing $\tilde{C}$.

In the ideal world, the decryption queries do not generate permutation evaluations, and the overlap between $\mathcal{M}_E$ and $\mathcal{M}_D$ impacts only the number of mock intermediate states. Remarking that $\ell(\tilde{\tau})$ is the number of superseded decryption paths, a good transcript induces $\mathcal{M}_E + \mathcal{N}$ distinct permutation evaluations, and $\mathcal{M}_D - \ell(\tilde{\tau})$ distinct random states $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$. Therefore,

$$\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right) = \frac{1}{(2^b)_{\mathcal{N}+\mathcal{M}_E}\,(2^b)^{\mathcal{M}_D-\ell(\tilde{\tau})}}\,\frac{1}{(2^k)^{\mu}}\,.$$

We therefore obtain

$$\frac{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right)}{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right)} = \frac{\left(2^b\right)_{\mathcal{N}+\mathcal{M}_E}\,(2^b)^{\mathcal{M}_D-\ell(\tilde{\tau})}}{(2^b)_{\mathcal{N}+\mathcal{M}_E+\mathcal{M}_D-\ell(\tilde{\tau})}} \geq 1\,. \tag{9}$$

Probability of $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ in the Ideal World. We do this evaluation in a query-wise fashion, upper bounding on-the-fly the probability that any fresh permutation evaluation or mock intermediate state triggers $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$. Permutation evaluations are considered in the order they occur, while mock intermediate states are considered at the end. Let $i \in [\![1, \mathcal{M}_E + \mathcal{N}]\!]$, and for an event $\mathbf{Evt}$, $\mathbf{Evt}[i]$ denotes the probability that $\mathbf{Evt}$ is set after i *fresh* permutation *evaluations* (coming either from permutation or encryption queries). Let $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[0]$ be $\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}}$, as this is the only event that can be set before any query from the distinguisher. Therefore, $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}} \wedge \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[\mathcal{M}_E + \mathcal{N}]$ denotes the event that one of the intermediate states generated after the interaction (i.e., by decryption queries that are not superseded) sets $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$.

Let $\mathbf{1}_{\mathtt{C}}[i]$ denote the indicator function equal to one if and only if the evaluation number i is *fresh* and made in the context of a construction query (here, necessarily an encryption query). Similarly, $\mathbf{1}_{\mathtt{P}}[i]$ equals one if and only if the evaluation number i is *fresh* and made from a permutation query. Note that whenever an encryption query (E, m, N, A, P, C, T) is made, there can exist earlier decryption queries of the form $(D, m, N, A, C, T_j, \perp)_j$. Define $\eta_{\mathtt{enc},\mathtt{dec}}[i]$ as follows:

- If evaluation i originates from a permutation query or an encryption query with a non-final path, then $\eta_{\mathtt{enc},\mathtt{dec}}[i] = 0$;
- Otherwise, let $(E, m, N, A, P, C, T) \in \tau$ be the associated encryption query, then $\eta_{\mathtt{enc},\mathtt{dec}}[i]$ counts the number of (necessarily earlier) decryption queries of the form $(D, m, N, A, C, T_j, \perp)$.

Since two encryption queries with the same user, nonce, associated data, but with different plaintexts cannot have the same ciphertexts, one single decryption query cannot contribute to increment two distinct $\eta_{\mathtt{enc},\mathtt{dec}}[i]$ and $\eta_{\mathtt{enc},\mathtt{dec}}[j]$. Therefore, we have $\sum_{i=1}^{\mathcal{M}_E+\mathcal{N}} \eta_{\mathtt{enc},\mathtt{dec}}[i] \leq Q_D$.

We break down the probability of $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ by using basic probability as follows:

1. $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[0]$, or $\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}}$ by definition;
2. $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i]$ for $i \in [\![1, \mathcal{M}_E + \mathcal{N}]\!]$, in more detail:
   (a) $\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1]$;
   (b) $\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i]$;[6]
   (c) $\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i]$;
   (d) $\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}[i]$;

[6] Note that, although $\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}$ involves a decryption query, any permutation evaluation that triggers this bad event during the interactive phase comes from an encryption query.

3. $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ after the interaction, which is equivalent to $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$ assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[\mathcal{M}_E + \mathcal{N}]$.

Case 2 can be set only with intermediate states from encryption queries while case 3 involves additionally intermediate states from decryption queries.

Case 1. Let us start with the bounding of case 1. We have

$$\mathbf{Pr}\left(\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[0]\right) = \mathbf{Pr}\left(\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}}\,. \tag{10}$$

Case 2. In the following, let $i \in [\![1, \mathcal{M}_E + \mathcal{N}]\!]$. We first focus on the conditioned $\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i]$ probability of case 2a. In order to set this event, the adversary must be able to guess one of the μ uniform random keys, either via a direct permutation call, or via a permutation evaluation made from an encryption query. In the second case, we can out of generosity *for this event only* assume that the adversary has full control on the input of the intermediate states. Moreover, each failed guess eliminates μ elements in $\{0,1\}^k$ from the set of candidate keys. Therefore,

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1]\right) &\leq \left(\mathbf{1}_{\mathtt{P}}[i] + \mathbf{1}_{\mathtt{C}}[i]\right)\frac{\mu}{2^k - \mu\left(\mathcal{M}_E + \mathcal{N}\right)} \\ &\leq \left(\mathbf{1}_{\mathtt{P}}[i] + \mathbf{1}_{\mathtt{C}}[i]\right)\frac{2\mu}{2^k}\,,\end{aligned} \tag{11}$$

where we used that $\mu\left(\mathcal{M} + \mathcal{N}\right) \leq 2^{k-1}$.

Regarding the conditioned $\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i]$ of case 2b, this event during the interactive phase can be set only during the generation of a final state of an encryption query. Those aforementioned states are sampled uniformly in a permutation-consistent way, added to the key, and truncated before output. The keys are uniformly random and hidden from the adversary, and there are by definition $\eta_{\mathtt{enc},\mathtt{dec}}[i]$ candidate tags to hit. Therefore,

$$\mathbf{Pr}\left(\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i]\right) \leq \eta_{\mathtt{enc},\mathtt{dec}}[i]\frac{1}{2^t}\,. \tag{12}$$

Then, we focus on the conditioned $\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}[i]$ probability in case 2c. This event can be set only by an evaluation from an encryption query. Let $\mathtt{path} \in \mathcal{P}$ be the associated encryption path. Assuming $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$, the state $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$ is sampled uniformly at random from a set of size at least $2^b - \mathcal{M}_E - \mathcal{N}$. Therefore, the probability that $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$ collides on its inner part with any other state $\mathtt{S}_{\mathrm{out}}[\mathtt{path}']$, modulo XORing key material or the domain separator bits (captured by the presence of δ and δ'), can be upper bounded by

$$\mathbf{1}_{\mathtt{C}}[i]\frac{9 \cdot 2^r \cdot \mathcal{M}_E}{2^b - \mathcal{M}_E - \mathcal{N}}\,,$$

where we used that $|\mathrm{ValidXor}\left(\mathtt{path}\right)| \leq 3$. Therefore,

$$\mathbf{Pr}\left(\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i]\right) \leq \mathbf{1}_{\mathtt{C}}[i]\frac{18\mathcal{M}_E}{2^c}\,, \tag{13}$$

where we used that $\mathcal{M}_E + \mathcal{N} \leq 2^{b-1}$.

Then, let us focus on the conditioned $\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}}[i]$ from case 2d. First, if the evaluation number i comes from a permutation query $(X, Y, \mathrm{d}) \in \tau$, then for every existing intermediate state $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ and $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$, we evaluate the probability that the query (X, Y, d) paired with the state sets $\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}$, and count the number of candidate states to be guessed:

- If $\mathrm{d} = \mathrm{fwd}$ (resp., $\mathrm{d} = \mathrm{inv}$), then the state Y (resp., X) is sampled uniformly at random from a set of size at least $2^b - \mathcal{M}_E - \mathcal{N}$, thus it collides with a $\mathtt{S}_{\mathrm{out}}[\mathtt{path}] \oplus \delta$

on its inner part with probability at most $\frac{3 \cdot 2^r}{2^b - \mathcal{M}_E - \mathcal{N}}$. There are at most $\mathcal{M}_E$ such states, thus we get a probability at most

$$\mathbf{1}_{\mathtt{P}}[i] \frac{6\mathcal{M}_E}{2^c} ,$$

where we used that $\mathcal{M}_E + \mathcal{N} \leq 2^{b-1}$. From now on, we consider the other cases, where the direction of the permutation aligns with the state to guess;

- If the state to guess is of the form $\mathtt{S}_{\text{out}}[(E, m, N, A, P, 1)]$, then conditioned on $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1]$, this state is sampled uniformly from a set of size at least $2^b - \mathcal{M}_E - \mathcal{N}$. The outer $b - t$ bits are completely hidden from the adversary. The rightmost t bits are added to $\lfloor K_m \rfloor_t$ before being returned as tag. Since the keys are random and hidden, access to those t bits is of no help for the adversary. The total number of distinct states $\mathtt{S}_{\text{out}}[(E, m, N, A, P, 1)]$ is upper bounded by Q_E. Therefore, this event is set with probability at most

$$\mathbf{1}_{\mathtt{P}}[i] \frac{Q_E}{2^b - \mathcal{M}_E - \mathcal{N}} \leq \mathbf{1}_{\mathtt{P}}[i] \frac{2Q_E}{2^b} ,$$

where we used that $\mathcal{M}_E + \mathcal{N} \leq 2^{b-1}$;

- Otherwise, the state to guess is an internal state (either the output of the first permutation evaluation, the input of the last permutation evaluation, or any input or output of a middle permutation evaluation). Since the adversary is allowed to repeat nonces, we can without loss of accuracy in the bounding assume that the states have their outer part set to a value of the adversary's choice, thus we consider equality on the inner part. Each of the states is sampled uniformly from a set of size at least $2^b - \mathcal{M}_E - \mathcal{N}$, and as long as $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1]$ holds, their inner part remains secret from the adversary. There are at most $3(\mathcal{M}_E - Q_E)$ states concerned (including potential future states), thus we obtain a probability at most

$$\mathbf{1}_{\mathtt{P}}[i] \frac{3(\mathcal{M}_E - Q_E) \cdot 2^r}{2^b - \mathcal{M}_E - \mathcal{N}} \leq \mathbf{1}_{\mathtt{P}}[i] \frac{6(\mathcal{M}_E - Q_E)}{2^c} ,$$

where we used that $\mathcal{M}_E + \mathcal{N} \leq 2^{b-1}$.

Then, if the evaluation number i comes from an encryption query and is associated to a path $\mathtt{path}$, in all cases the bad event can be triggered only by the randomness of $\mathtt{S}_{\text{out}}[\mathtt{path}]$. There are at most 3 different possible values for δ, and $\mathtt{S}_{\text{out}}[\mathtt{path}]$ is sampled uniformly at random from a set of size at least $2^b - \mathcal{M}_E - \mathcal{N}$. Therefore, this event is set with probability at most

$$\mathbf{1}_{\mathtt{C}}[i] \frac{6\mathcal{N}}{2^c} ,$$

where we used that $\mathcal{M}_E + \mathcal{N} \leq 2^{b-1}$. Therefore,

$$\begin{aligned} \mathbf{Pr}\left(\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}[i] \wedge \neg\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}[i]\right) \\ \leq \mathbf{1}_{\mathtt{P}}[i] \frac{12\mathcal{M}_E}{2^c} + \mathbf{1}_{\mathtt{C}}[i] \frac{6\mathcal{N}}{2^c} . \quad (14) \end{aligned}$$

Case 3. Finally, we can focus on the conditioned bad event in case 3. Because $\neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[\mathcal{M}_E + \mathcal{N}]$ holds, then in order to be set, the bad event necessarily involves an intermediate state $\mathtt{S}_{\text{out}}[\mathtt{path}]$, for $\mathtt{path}$ a decryption path not superseded. The inner part of these states are sampled uniformly at random, with at most $\mathcal{M}_D$ such states in total. We evaluate all sub-events as follows:

- $\mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}}$: this event cannot be set after the interaction;
- $\mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}}$: this event is set with probability at most $\frac{2\mu\mathcal{M}_D}{2^k}$, where we used that $\mu\,(\mathcal{N}+\mathcal{M}) \leq 2^{k-1}$;
- $\mathbf{ColS}^{\mathbf{m}}_{\mathbf{a}}$: this event is set with probability at most $\frac{9\mathcal{M}_D(\mathcal{M}_E+\mathcal{M}_D)}{2^c}$;
- $\mathbf{GueS}^{\mathbf{m}}_{\mathbf{a}}$: this event is set with probability at most $\frac{6\mathcal{M}_D\mathcal{N}}{2^c}$;
- $\mathbf{Dec}^{\mathbf{m}}_{\mathbf{a}}$: this event is set with probability at most $\frac{Q_D}{2^t}$.

Therefore,

$$\mathbf{Pr}\left(\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}} \mid \neg\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}[\mathcal{M}_E+\mathcal{N}]\right) \leq \frac{2\mu\mathcal{M}_D}{2^k} + \frac{9\mathcal{M}_D(\mathcal{M}_E+\mathcal{M}_D)}{2^c} + \frac{6\mathcal{M}_D\mathcal{N}}{2^c} + \frac{Q_D}{2^t}\,. \quad (15)$$

Conclusion. By combining the conditioned probabilities of (10) to (15), we obtain

$$\mathbf{Pr}\left(\mathbf{BAD}^{\mathbf{m}}_{\mathbf{a}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\,(\mathcal{M}+\mathcal{N})}{2^k} + \frac{18\mathcal{M}\,(\mathcal{M}+\mathcal{N})}{2^c} + \frac{2Q_D}{2^t}\,.$$

We obtained an upper bound for the probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (9)), we conclude. □

6.3 Proof of Theorem 3

Authenticity in the nonce-misuse resilience setting is implied by authenticity in the nonce-misuse setting (i.e., Theorem 2). This proof is therefore dedicated to confidentiality. This proof will re-use a significant part of the notation defined in Section 6.2, and we will explicitly highlight the adaptations made here.

Let $\mathcal{A}$ be an adversary that makes at most $\mathcal{N}$ permutation queries, and Q_E encryption queries of at most $\mathcal{M}_E$ blocks, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}^{\mu\text{-mr-conf}}_{\text{Ascon}}(\mathcal{A})$. The adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\mathbf{Enc}^p_{K_m}, \mathbf{Enc}^p_{K_m}\right)^{\mu}_{m=1}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\mathbf{Enc}^p_{K_m}, \$_m\right)^{\mu}_{m=1}, p^{\pm}\right]$. $\mathcal{A}$ may misuse nonces with the queries to $\mathcal{O}_{1,m}$, but not with $\mathcal{O}_{2,m}$. We will refer to a query to $\mathcal{O}_{2,m}$ as a *challenge* query, and to $\mathcal{O}_{1,m}$ as a *learning* query. Let $\mathcal{M}_{E,C}$ be the data complexity of challenge queries, and $\mathcal{M}_{E,L}$ the one of learning queries, so that $\mathcal{M}_E = \mathcal{M}_{E,C} + \mathcal{M}_{E,L}$.

Transcript Notation. We define below notation for the transcript τ:

- A forward (resp., inverse) permutation query with input X and output Y generates the transcript element (X, Y, fwd) (resp., (X, Y, inv));
- A challenge encryption query with user m, input (N, A, P), and output (C, T) generates the transcript element (E_C, m, N, A, P, C, T);
- A learning encryption query with user m, input (N, A, P), and output (C, T) generates the transcript element (E_L, m, N, A, P, C, T).

Paths and Intermediate States. We will re-use the path notation from Section 6.2. This time, all construction queries are encryption queries, so that the paths take the form $\mathtt{path} = (O, m, N, A, P, f) \in \mathcal{P}$, where $O = E_C$ if the associated query is a challenge query, otherwise $O = E_L$. Thanks to the fact that the queries to $\mathcal{O}_{1,m}$ and $\mathcal{O}_{2,m}$ do not have overlapping nonces, a challenge path cannot be the parent of a learning path, and vice-versa. Because the queries to the challenge oracle are nonce-respecting, each challenge path can have at most one child. We define the dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$ similarly to Section 6.2. The labels of the dictionaries are in $\mathcal{P}$ in the real world, or the set of learning paths in the ideal world.

Mock Intermediate States. Again, our approach will be to establish an extended transcript that releases the intermediate states associated to all construction queries. Therefore, we define in the following a procedure to generate mock intermediate states in the ideal world for $\mathtt{S}_{\text{in}}[\mathtt{path}]$ and $\mathtt{S}_{\text{out}}[\mathtt{path}]$ for any challenge path $\mathtt{path}$. The sampling procedure, taking place at the end of the interaction, operates as follows:

- Sample $\mathtt{S}_{\text{in}}[\mathtt{path}]$, for all $\mathtt{path} = (E_C, m, N, A, P, f)$ as follows:
  - If $P = A = \emptyset$, then necessarily $f = 0$ and $\mathtt{S}_{\text{in}}[\mathtt{path}] \leftarrow IV \| K_m \| N$;
  - Else, if $P = \emptyset$, then $\mathtt{S}_{\text{in}}[\mathtt{path}] \xleftarrow{\$} \{0,1\}^b$;
  - Else, let $l \in \mathbb{N}$ be the length of P, find the (unique) encryption query with user m and nonce N, get the ciphertext block number l (name it C_l), and sample $Z \xleftarrow{\$} \{0,1\}^c$. Then: $\mathtt{S}_{\text{in}}[\mathtt{path}] \leftarrow C_l \| Z$;
- Then, sample $\mathtt{S}_{\text{out}}[\mathtt{path}]$, for all $\mathtt{path} = (E_C, m, N, A, P, f)$ as follows:
  - If $f = 1$, the existence of such a path means that there exists $(E_C, m, N, A, P, C, T) \in \tau$. Then, sample $Z \xleftarrow{\$} \{0,1\}^{b-t}$, and: $\mathtt{S}_{\text{out}}[\mathtt{path}] \leftarrow Z \| (\lfloor K_m \rfloor_t \oplus T)$;
  - Otherwise, let $\mathtt{pathC} = (m, N, A', P', f')$ be the (necessarily unique) child of $\mathtt{path}$, and let $B \in \{0,1\}^r$ be the last block of $A' \| P'$, then:

$$\mathtt{S}_{\text{out}}[\mathtt{path}] \leftarrow (B \| 0^c) \oplus \text{XorState}\,(\mathtt{path}, \mathtt{pathC}) \oplus \mathtt{S}_{\text{in}}[\mathtt{pathC}]\,.$$

All of these states are generated *after* the interactive phase. However, some of them might use randomness from the non-interactive phase (e.g., the $\mathtt{S}_{\text{in}}[(E_C, m, N, \emptyset, \emptyset, 0)]$ are fixed by the user key K_m, and the $\mathtt{S}_{\text{out}}[(E_C, m, N, A, P, 1)]$ have their rightmost t bits set by the encryption query output, along with the key K_m).

Extended Transcript. Now that the dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$ are defined for all labels in $\mathcal{P}$ in both worlds, we can define the extended transcript $\tilde{\tau}$ built from τ by replacing adding elements as follows:

- All permutation queries (X, Y, d) are kept untouched;
- An encryption query tuple $(O, m, N, A, P, C, T) \in \tau$ is now followed by tuples of the form $(\mathtt{path}, \mathtt{S}_{\text{in}}[\mathtt{path}], \mathtt{S}_{\text{out}}[\mathtt{path}])$ for all $\mathtt{path}$ generated by (m, N, A, P). If some paths repeat (due to the nonce-misuse setting), then the duplicates are removed from the transcript;
- At the end of $\tilde{\tau}$, a tuple containing the keys $(K_1, \ldots, K_\mu)$ is added.

Summarizing, from $\tilde{\tau}$ we can reconstruct the two dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$, the set $\mathcal{P}$, the list of all permutation queries (X, Y, d), and the user's keys. This transcript is released at the end of the interaction, right before the distinguisher outputs its decision bit.

Bad Events. We will re-use the bad events defined in Section 6.2, except **Dec**, which does not apply there. In order to completely inherit the notation from this section, let us define $\mathcal{P}_{\mathrm{S}} = \emptyset$. The bad events are as follows:

$$\begin{aligned}
&\mathbf{ColK_c^{mr}} : \mathbf{ColK_a^m} \text{ (of Section 6.2)}\,; \quad && \mathbf{GueK_c^{mr}} : \mathbf{GueK_a^m} \text{ (of Section 6.2)}\,;\\
&\mathbf{ColS_c^{mr}} : \mathbf{ColS_a^m} \text{ (of Section 6.2)}\,; && \mathbf{GueS_c^{mr}} : \mathbf{GueS_a^m} \text{ (of Section 6.2)}\,;\\
&\mathbf{BAD_c^{mr}} : \mathbf{ColK_c^{mr}} \vee \mathbf{GueK_c^{mr}} \vee \mathbf{ColS_c^{mr}} \vee \mathbf{GueS_c^{mr}}\,.
\end{aligned}$$

Probability of Good Transcripts. As in Section 6.2, the absence of $\mathbf{BAD_c^{mr}}$ guarantees that the intermediate states generated in the ideal world are consistent with the structure of the mode Ascon-AE. Therefore, every transcript which is reachable in the real world is also reachable in the ideal world, and vice-versa. Let $\tilde{\tau}$ be a transcript that does not set $\mathbf{BAD_c^{mr}}$. In the real world, this transcript induces $\mathcal{N} + \mathcal{M}_{E,L} + \mathcal{M}_{E,C}$ permutation calls and μ keys, thus

$$\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right) = \frac{1}{(2^b)_{\mathcal{N}+\mathcal{M}_{E,L}+\mathcal{M}_{E,C}}} \frac{1}{(2^k)^{\mu}}\,.$$

In the ideal world, this transcript induces $\mathcal{N} + \mathcal{M}_{E,L}$ permutation calls, $\mathcal{M}_{E,C}$ random values, and μ keys. Therefore,

$$\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right) = \frac{1}{(2^b)_{\mathcal{N}+\mathcal{M}_{E,L}}\,(2^b)^{\mathcal{M}_{E,C}}} \frac{1}{(2^k)^{\mu}}\,.$$

Therefore,

$$\frac{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right)}{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right)} = \frac{(2^b)_{\mathcal{N}+\mathcal{M}_{E,L}}\,(2^b)^{\mathcal{M}_{E,C}}}{(2^b)_{\mathcal{N}+\mathcal{M}_{E,L}+\mathcal{M}_{E,C}}} \geq 1\,. \tag{16}$$

Probability of $\mathbf{BAD_c^{mr}}$ in the Ideal World. This will be done in a query-wise fashion, similarly to Section 6.2. In the ideal world, the only encryption queries that trigger permutation evaluations are learning queries. Let $i \in [\![0, \mathcal{M}_{E,L} + \mathcal{N}]\!]$, and denote by $\mathbf{BAD_c^{mr}}[i]$ the probability that $\mathbf{BAD_c^{mr}}$ is set after i *fresh* permutation *evaluations* (coming either from permutation or learning construction queries). Here, as before, $\mathbf{BAD_c^{mr}}[0] = \mathbf{ColK_c^{mr}}$.

By basic probability, we have

$$\begin{aligned}
\mathbf{Pr}\left(\mathbf{BAD_c^{mr}}\right) \leq\ & \mathbf{Pr}\left(\mathbf{BAD_c^{mr}}[\mathcal{M}_{E,L} + \mathcal{N}]\right)\\
&+ \mathbf{Pr}\left(\mathbf{BAD_c^{mr}} \mid \neg\mathbf{BAD_c^{mr}}[\mathcal{M}_{E,L} + \mathcal{N}]\right)\,.
\end{aligned} \tag{17}$$

We remark that the bounding of the query-wise event $\mathbf{BAD_c^{mr}}[\mathcal{M}_{E,L} + \mathcal{N}]$ can be performed the same way as in the proof from Section 6.2, the differences being that (i) at most $\mathcal{M}_{E,L}$ permutation evaluations from encryption queries are made during the interactive phase (as opposed to $\mathcal{M}_E$ in Section 6.2), and (ii) the bad event **Dec** does not apply here. Therefore,

$$\begin{aligned}
&\mathbf{Pr}\left(\mathbf{BAD_c^{mr}}[\mathcal{M}_{E,L} + \mathcal{N}]\right)\\
&\qquad \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{M}_{E,L} + \mathcal{N}\right)}{2^k} + \frac{18\left(\mathcal{M}_{E,L}\right)^2}{2^c} + \frac{18\mathcal{M}_{E,L}\mathcal{N}}{2^c}\,.
\end{aligned} \tag{18}$$

The bad events in the second term appearing in (17) can also be upper bounded the same way, noticing again that this phase involves $\mathcal{M}_{E,C}$ fresh intermediate states (and not $\mathcal{M}_D$). This time, the intermediate states have their entire b bits generated randomly, but since

the bad events are defined only on the inner part, we can use the same bounding technique. We obtain

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{BAD}_{\mathbf{c}}^{\mathbf{mr}} \mid \neg\mathbf{BAD}_{\mathbf{c}}^{\mathbf{mr}}[\mathcal{M}_{E,L}+\mathcal{N}]\right)\\ \leq \frac{2\mu\mathcal{M}_{E,C}}{2^k} + \frac{9\mathcal{M}_{E,C}(\mathcal{M}_{E,L}+\mathcal{M}_{E,C})}{2^c} + \frac{6\mathcal{M}_{E,C}\mathcal{N}}{2^c}\,. \quad (19)\end{aligned}$$

Finally, combining (18) and (19) into (17), we obtain

$$\mathbf{Pr}\left(\mathbf{BAD}_{\mathbf{c}}^{\mathbf{mr}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{M}_E+\mathcal{N}\right)}{2^k} + \frac{18\mathcal{M}_E\left(\mathcal{M}_E+\mathcal{N}\right)}{2^c}\,.$$

We obtained an upper bound for probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (16)), we conclude. □

6.4 Proof of Theorem 4

We consider authenticity in Section 6.4.1 and confidentiality in Section 6.4.2.

6.4.1 Authenticity

Let $\mathcal{A}$ be an adversary that makes at most $\mathcal{N}$ permutation queries, Q_E encryption queries of at most $\mathcal{M}_E$ blocks, and Q_D decryption queries of at most $\mathcal{M}_D$ blocks, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}^{\mu\text{-lr-auth}}_{\text{Ascon},\mathcal{L}}(\mathcal{A})$, for any set of leakage functions that do not leak any information about the two outer permutation calls during the initialization and finalization phases. We therefore assume maximal leakage, so that the leakage function leaks all inner permutation calls. The adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \mathbf{Dec}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L, \bot\right)_{m=1}^{\mu}, p^{\pm}\right]$.

The adversary must be nonce-respecting with its queries to $\mathcal{O}_{2,m}$. The oracles $\mathcal{O}_{1,m}$ and $\mathcal{O}_{3,m}$ will be referred to as the *leaky oracles*, while the oracles $\mathcal{O}_{2,m}$ and $\mathcal{O}_{4,m}$ will be referred to as the *challenge oracles*. Although $\mathcal{O}_{2,m}$ does not produce real-or-random strings, we still categorize it as a challenge oracle because the nonces used with $\mathcal{O}_{2,m}$ can be re-used with the decryption oracle $\mathcal{O}_{4,m}$. To accurately track the adversarial resources, we refine the online complexity $(Q, \mathcal{M})$ as follows: it is split into the queries to the challenge oracles $(Q_C, \mathcal{M}_C)$ and queries to the leaky oracles $(Q_L, \mathcal{M}_L)$. These are further divided into encryption and decryption queries, with the symbols D or E prepended to the subscript. For example, $(Q_{E,L}, \mathcal{M}_{E,L})$ represents the online complexity of queries made to the leaky encryption oracle.

Transcript Notation. We define below a notation for the transcript τ:

- A forward (resp., inverse) permutation query with input X and output Y generates the transcript element (X, Y, fwd) (resp., (X, Y, inv));
- A leaky query with user m generates the following transcript elements:
  - If the query is an encryption query with input (N, A, P), and output (C, T), the element (E_L, m, N, A, P, C, T) is added to the transcript;
  - If the query is a decryption query with input (N, A, C, T), and output $\tilde{P}$, the element $(D_L, m, N, A, C, T, \tilde{P})$ is added to the transcript;

– An element (X, Y, cons), for all inner permutation evaluations made with input X and output Y from the construction. In other words, all permutation evaluations are added to the transcript, *except the two outer ones*;

- A query to the challenge encryption oracle with user m, input (N, A, P), and output (C, T), generates the transcript element (E_C, m, N, A, P, C, T);
- A query to the challenge decryption oracle with user m, input (N, A, C, T), and output $\tilde{P}$, generates the transcript element $(D_C, m, N, A, C, T, \tilde{P})$.

Paths and Intermediate States. We adapt the path notation from Section 6.2 as follows:

- The paths generated by the challenge queries inherit the inductive path definition from Section 6.2, with one administrative modification: the very first element of the path is adjusted to distinguish it as a challenge path. In detail, a challenge path takes the form $\texttt{path} = (O, m, N, A, B, f)$, where $O = E_C$ if the associated query is an encryption query, and $O = D_C$ if the associated query is a decryption query. Denote by $\mathcal{P}_C$ the set of paths generated by this procedure, and let $\mathcal{P}_{\mathrm{S}}$ be the set of decryption paths in $\mathcal{P}_C$ that are superseded;
- A leaky query $(O, m, N, A, B, -) \in \tau$ with $O \in \{E_L, D_L\}$ generates exactly two paths: $(O, m, N, \emptyset, \emptyset, 0)$ and $(O, m, N, A, B, 1)$. All other intermediate states are given to the adversary and treated as direct permutation queries in the transcript. Let $\mathcal{P}_L$ be the set of paths generated by leaky construction queries.

Let $\mathcal{P} = \mathcal{P}_L \cup \mathcal{P}_C$ be the set of all paths generated by construction queries. We define the dictionaries $\mathtt{S}_{\mathrm{in}}$ and $\mathtt{S}_{\mathrm{out}}$ as in Section 6.2, with labels in $\mathcal{P}$ in the real world. In the ideal world, however, the challenge decryption paths are (not yet) present. We stress that the leaked intermediate states (X, Y, cons) are absent in the set of paths $\mathcal{P}$.

Mock Intermediate States. Again, we aim to establish an extended transcript that releases the intermediate states, hence we need to specify a procedure to sample $\mathtt{S}_{\mathrm{in}}[\mathtt{path}]$ and $\mathtt{S}_{\mathrm{out}}[\mathtt{path}]$ for any challenge decryption path $\mathtt{path}$. The procedure is the same as the one defined in Section 6.2, and we can safely ignore leaky paths, since they do not have overlapping nonces.

Extended Transcript. Now that the dictionaries $\mathtt{S}_{\mathrm{in}}$ and $\mathtt{S}_{\mathrm{out}}$ are defined for all keys in $\mathcal{P}$ in both worlds, we can define the extended transcript $\tilde{\tau}$ built from τ as follows:

- All permutation evaluations from permutation queries or from leaky construction queries (X, Y, d) are kept untouched, except that if a tuple (X, Y) repeats, then it is removed from $\tilde{\tau}$;
- A challenge construction query $(O, m, N, A, P, C, T) \in \tau$ is now followed by tuples of the form $(\mathtt{path}, \mathtt{S}_{\mathrm{in}}[\mathtt{path}], \mathtt{S}_{\mathrm{out}}[\mathtt{path}])$ for all $\mathtt{path}$s descendant of the aforementioned query. If some paths repeat (due to the nonce-misuse setting, or if a decryption path is superseded by a challenge encryption path), then the duplicates are removed from the transcript;
- A leaky construction query $(O, m, N, A, P, C, T) \in \tau$ is now followed by tuples of the form $(\mathtt{path}, \mathtt{S}_{\mathrm{in}}[\mathtt{path}], \mathtt{S}_{\mathrm{out}}[\mathtt{path}])$, for the two elements $\mathtt{path}$ generated by the aforementioned query. Again, if some paths are redundant, then the duplicates are removed from the transcript;
- At the end of $\tilde{\tau}$, a tuple containing the keys $(K_1, \ldots, K_\mu)$ is added.

This transcript is released at the end of the interaction, right before the distinguisher outputs its decision bit.

Bad Events. We introduce the following bad events:

$$\begin{aligned}
\mathbf{ColK}^{\mathbf{lr}}_{\mathbf{a}} &: \mathbf{ColK}^{\mathbf{m}}_{\mathbf{a}} \text{ (of Section 6.2)}\,; \quad \mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}} : \mathbf{GueK}^{\mathbf{m}}_{\mathbf{a}} \text{ (of Section 6.2)}\,;\\
\mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}} &: \exists (D_C, m, N, A, C, T, \tilde{P}) \in \tilde{\tau} \text{ such that } \lfloor \mathtt{S}_{\mathrm{out}}[(D_C, m, N, A, C, 1)] \rfloor_t \oplus \lfloor K_m \rfloor_t = T, \text{ or}\\
&\quad \exists \mathtt{path} \in \mathcal{P} \text{ superseding } (D_C, m, N, A, C, 1) \text{ with } \lfloor \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \rfloor_t \oplus \lfloor K_m \rfloor_t = T\,;\\
\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}} &: \exists^{\neq} \mathtt{path} \in \mathcal{P}_C \setminus \mathcal{P}_{\mathrm{S}}, \mathtt{path}' \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{S}}\\
&\quad \text{such that } \mathtt{S}_{\mathrm{in}}[\mathtt{path}] \stackrel{c}{=} \mathtt{S}_{\mathrm{in}}[\mathtt{path}'] \text{ or } \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \stackrel{c}{=} \mathtt{S}_{\mathrm{out}}[\mathtt{path}']\,;\\
\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}} &: \exists (X, Y, \mathrm{d}) \in \tilde{\tau}, \mathtt{path} \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{S}} \text{ such that } X \stackrel{c}{=} \mathtt{S}_{\mathrm{in}}[\mathtt{path}] \text{ or } Y \stackrel{c}{=} \mathtt{S}_{\mathrm{out}}[\mathtt{path}]\,;\\
\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}} &: \mathbf{ColK}^{\mathbf{lr}}_{\mathbf{a}} \vee \mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}} \vee \mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}} \vee \mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}} \vee \mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}}\,.
\end{aligned}$$

Compared to the bad events from Section 6.2, $\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}$ now only considers collisions involving a challenge intermediate state. Moreover, $\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}$ has been adjusted to account for the fact that not all intermediate states are added to the set $\mathcal{P}$. Finally, $\mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}}$ accounts for the fact that the decryption queries are clearly distinguished as challenge queries, though this adjustment is purely administrative.

Probability of Good Transcripts. The absence of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$ prevents the ideal world from generating states which do not adhere to the structure of the Ascon-AE mode. In addition, the absence of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$ (or, more precisely, of $\mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}}$) in the real world prevents the challenge decryption oracle from returning a string $\tilde{P}$ different from $\perp$. Therefore, every good transcript which is reachable in the real world is also reachable in the ideal world, and vice-versa. Let $\tilde{\tau}$ be a good transcript. Such a transcript induces a certain number of permutation evaluations from the queries to the leaky oracles and the permutation queries. These evaluations do not overlap with evaluations from the challenge oracles, and the count is identical in both the real and ideal worlds. Moreover, in the real world, the challenge queries induce $\mathcal{M}_{D,C} + \mathcal{M}_{E,C} - |\mathcal{P}_{\mathrm{S}}|$ additional permutation evaluations, while in the ideal world this induces $\mathcal{M}_{E,C}$ additional permutation evaluations and $\mathcal{M}_{D,C} - |\mathcal{P}_{\mathrm{S}}|$ random b-bit states. Therefore, similarly to Section 6.2, we have

$$\frac{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right)}{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right)} \geq 1\,. \tag{20}$$

Probability of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$ in the Ideal World. We will again use a query-wise approach to evaluate the probability to set $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$. The number of permutation evaluations done during the interactive phase is at most $\mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}$. Let $i \in [\![0, \mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]\!]$. We will re-use the notation $\eta_{\mathsf{enc},\mathsf{dec}}[i]$ from Section 6.2. Since part of the leaky evaluations is treated the same way as permutation queries, we refine the indicator functions $\mathbf{1}_{\mathtt{C}}[i]$ and $\mathbf{1}_{\mathtt{P}}[i]$ into the following functions:

- $\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]$ is equal to 1 if and only if the evaluation number i is fresh, and (i) either the evaluation is from a direct permutation query, or (ii) the evaluation comes from a leaky construction evaluation. `CL` stands for "construction leaky". The total number of is setting this function to 1 is the number of tuples (X, Y, d) in the extended transcript, thus at most $\mathcal{N} + \mathcal{M}_L - Q_L$;
- $\mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]$ is equal to 1 if and only if the evaluation number i is fresh, and comes from a construction evaluation during the leftmost or rightmost permutation evaluation. $\mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]$ is itself refined into $\mathbf{1}_{\mathtt{C},\mathtt{KBI}}[i]$ and $\mathbf{1}_{\mathtt{C},\mathtt{KBF}}[i]$, for respectively the initial and the final evaluation. Each of these functions is set to 1 by at most $Q_E + Q_{D,L}$ different indexes. `KBI` and `KBF` stand for respectively "initial key blinding" and "final key blinding";

- $\mathbf{1}_{\mathtt{CH}}[i]$ is equal to 1 if and only if the permutation evaluation number i is fresh, and originates from an internal state generated during a challenge permutation query (excluding thus the key blindings). Thanks to the nonce-respecting setting, the number of is that set this function to 1 is at most $\mathcal{M}_{E,C} - 2Q_{E,C}$.

We will derive two distinct bounds, with the second involving an additional auxiliary event, which helps to manage inner collisions. Depending on the adversarial resources, the tighter of the two bounds will apply.

Probability of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$ in the Ideal World, First Bound. In this bounding, we evaluate $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}} := \mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$, without an additional auxiliary event. We break down the probability of $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}$ by using basic probability as follows:

1. $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[0]$, or $\mathbf{ColK}^{\mathbf{lr}}_{\mathbf{a}}$ by definition;
2. $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i]$ for $i \in [\![1, \mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]\!]$, in more detail:
   (a) $\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1]$;
   (b) $\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]$;
   (c) $\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]$;
   (d) $\mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}}[i]$, assuming $\neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]$;
3. $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}$ at the end of the interaction, which is equivalent to $\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}$ assuming $\neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[\mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]$.

Cases 1, 2a, 2d, and 3 can be upper bounded the same way as in Section 6.2, so that

$$\mathbf{Pr}\left(\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[0]\right) \leq \frac{\mu(\mu-1)}{2^{k+1}}\,, \tag{21}$$

$$\mathbf{Pr}\left(\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1]\right) \leq (\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] + \mathbf{1}_{\mathtt{CH}}[i] + \mathbf{1}_{\mathtt{C},\mathtt{KBF}}[i])\,\frac{2\mu}{2^k}\,, \tag{22}$$

$$\mathbf{Pr}\left(\mathbf{Dec}^{\mathbf{lr}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]\right) \leq \eta_{\mathtt{enc},\mathtt{dec}}[i]\frac{1}{2^t}\,, \tag{23}$$

$$\mathbf{Pr}\left(\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}} \mid \neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[\mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]\right) \leq \frac{2\mu\mathcal{M}_{D,C}}{2^k} + \frac{Q_{D,C}}{2^t} + \frac{2\mathcal{M}_{D,C}\,(\mathcal{M}+\mathcal{N})}{2^c}\,. \tag{24}$$

where we used that $\mu\,(\mathcal{N}+\mathcal{M}) \leq 2^{k-1}$. Now, we have two cases left.

Case 2b. We focus on the conditioned $\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]$. If the evaluation is from a permutation query or a leaky internal state evaluation, it can target either an input/output of a leftmost/rightmost permutation call, or a challenge intermediate state. For challenge intermediate states, using that the inner part of challenge intermediate states is secret and hidden from the adversary, we obtain a probability of at most

$$\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{4\mathcal{M}_{E,C}}{2^c}\,,$$

where we used that $\mathcal{M}+\mathcal{N} \leq 2^{b-1}$. For the key blinding input/outputs, the adversary has, in the best case, access to the input of the evaluation before the key additions and the output after the key additions. It therefore remains to guess the state after key addition.[7]

[7] For the output of the rightmost key blinding, we take a lossy step here.

There are 3 different places where the key blinding is applied,[8] thus in total at most $3(Q_E + Q_{D,L})$ states to be guessed. Therefore, this event is set with a probability of at most

$$\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{6\left(Q_E + Q_{D,L}\right)}{2^k}\,,$$

where we used that $Q\left(\mathcal{M}+\mathcal{N}\right) \leq 2^{k-1}$. Else, if the evaluation i is from an internal state evaluation due to a challenge query, we obtain a probability of at most

$$\mathbf{1}_{\mathtt{CH}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_L\right)}{2^c}\,,$$

where we used that $\mathcal{M}+\mathcal{N} \leq 2^{b-1}$. Else, if the evaluation i is from a construction query during the leftmost or rightmost evaluation, we obtain a probability of at most

$$\mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_L\right)}{2^k}\,,$$

where we used that $Q\left(\mathcal{M}+\mathcal{N}\right) \leq \mu 2^{k-1}$. Therefore,

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]\right) &\leq \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{4\mathcal{M}_{E,C}}{2^c} \\ &+ \mathbf{1}_{\mathtt{CH}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_L\right)}{2^c} + \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{6\left(Q_E + Q_{D,L}\right)}{2^k} + \mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_L\right)}{2^k}\,. \quad (25)\end{aligned}$$

Case 2c. We focus on the conditioned $\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}[i]$. With this event, only collisions with a challenge intermediate state matter. Those states have a secret inner part, and can be evaluated similarly to the conditioned $\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]$. Therefore,

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{ColKS}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i]\right) & \\ \leq \mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]\frac{4\mathcal{M}_{E,C}}{2^c} + \mathbf{1}_{\mathtt{CH}}[i]\left(\frac{6Q_L}{2^c} + \frac{4\mathcal{M}_{E,C}}{2^c}\right)\,, & \quad (26)\end{aligned}$$

where we used $\mathcal{M}+\mathcal{N} \leq 2^{b-1}$.

Combining (21) to (26), and simplifying the bounds with constant factor losses, we obtain

$$\mathbf{Pr}\left(\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{14Q\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{2Q_D}{2^t} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c}\,. \quad (27)$$

Probability of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$ in the Ideal World, Second Bound. We introduce the following auxiliary bad event $\mathbf{Inner}^{\mathbf{lr}}_{\mathbf{a}}$, set whenever there exists $i \in [\![1, \mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]\!]$ such that the output of the evaluation i collides on its inner part with a prior permutation evaluation input or output.

Let $\mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{a}} = \mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}} \vee \mathbf{Inner}^{\mathbf{lr}}_{\mathbf{a}}$, we will evaluate the probability of $\mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{a}}$. Let

[8] Note that due to the absence of $\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]$, the event can never be set with the input of the leftmost permutation call.

$i \in [\![0, \mathcal{M}_E + \mathcal{M}_{D,L} + \mathcal{N}]\!]$. We introduce the following random variables:

$$\Theta_{\mathrm{KBI}}[i] = \max_{Z \in \{0,1\}^{c-k}} \Big| \big\{ \mathtt{path} = (m, N, \emptyset, \emptyset, 0) \in \mathcal{P}[1:i] \mid \exists p \, (\mathtt{S}_{\mathrm{in}}[\mathtt{path}]) \text{ in forward direction such that } \mathtt{S}_{\mathrm{out}}[\mathtt{path}][c+1:c-k] = Z \big\} \Big| ,$$

$$\Theta_{\mathrm{KBF}}[i] = \max_{Z \in \{0,1\}^{c-k}} \Big| \big\{ (X, Y, \mathrm{d}) \in \tilde{\tau}[1:i] \mid \mathrm{d} \in \{\mathrm{fwd}, \mathrm{cons}\} \text{ and } \lfloor Y \rfloor_{c-k} = Z \big\} \cup \big\{ \mathtt{path} = (m, N, \emptyset, \emptyset, 0) \in \mathcal{P}[1:i] \mid \exists p \, (\mathtt{S}_{\mathrm{in}}[\mathtt{path}]) \text{ in forward direction such that } \lfloor \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \rfloor_{c-k} = Z \big\} \Big| ,$$

where we abuse notation with $\mathcal{P}[1:i]$ to denote the set of path descendants of the queries made up to and including the evaluation number i.

Moreover, let Θ_{KBI} and Θ_{KBF} denote respectively $\Theta_{\mathrm{KBI}}[\mathcal{N}+\mathcal{M}_E+\mathcal{M}_{D,L}]$ and $\Theta_{\mathrm{KBF}}[\mathcal{N}+\mathcal{M}_E+\mathcal{M}_{D,L}]$. The variable Θ_{KBI} counts the maximal size of multicollisions on the middle $c-k$ bits made from the first permutation evaluation in the construction queries. The variable Θ_{KBF} counts the maximal size of multicollisions on the lowest $c-k$ bits, over (almost) all permutation evaluations made in the forward direction. It is constructed from two sets: one that gathers the states generated from challenge queries or construction queries with a single plaintext/ciphertext block, and another one that gathers forward permutation queries and leaky permutation evaluations. All forward permutation evaluations are included since they can potentially be presented as input to the final permutation evaluation. Looking ahead, since $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}$ is included in the bad events, this means that, as long as $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}$ does not occur, the internal states generated from construction queries are only from forward permutation evaluations, and we can use Θ_{KBI} and Θ_{KBF} to tame multicollisions. We thus have

$$\mathbf{Ex}\left(\Theta_{\mathrm{KBI}}\right) \leq \mathrm{mucol}\left(Q, 2^{c-k}\right) , \qquad \mathbf{Ex}\left(\Theta_{\mathrm{KBF}}\right) \leq \mathrm{mucol}\left(\mathcal{M}+\mathcal{N}, 2^{c-k}\right) . \tag{28}$$

We will evaluate the probability of $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}$. Again, we break down the probability of $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}$ using basic probabilities as follows:

1. $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[0]$: same bounding as (21);
2. $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i]$, for all $i \in [\![1, \mathcal{N} + \mathcal{M}_E + \mathcal{M}_{D,L}]\!]$, we evaluate:
   (a) $\mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i]$, assuming $\neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1]$: same bounding as (22);
   (b) $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i]$, assuming $\neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1] \wedge \neg\mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{ColKS}_{\mathbf{a}}^{\mathbf{lr}}[i]$: the reasoning and bounding is done later in (29);
   (c) $\mathbf{GueS}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1] \wedge \neg\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i]$: the reasoning and bounding is done later in (31);
   (d) $\mathbf{ColS}_{\mathbf{a}}^{\mathbf{lr}}[i]$, assuming $\neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1] \wedge \neg\mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{GueS}_{\mathbf{a}}^{\mathbf{lr}}[i]$: same bounding as (26);
   (e) $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{lr}}[i]$, assuming $\neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1] \wedge \neg\mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{ColS}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg\mathbf{GueS}_{\mathbf{a}}^{\mathbf{lr}}[i]$: same bounding as (23);
3. $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}$ at the end of the interaction, which is equivalent to $\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}$ assuming $\neg\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[\mathcal{N} + \mathcal{M}_E + \mathcal{M}_{D,L}]$: $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}$ cannot be set at the end of the interaction, and the same bound as (24) can be derived.

We are left with two cases.

Case 2b. We start to evaluate the conditioned $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i]$. This corresponds to the probability that a b-bit string generated in a permutation-consistent way collides on its

inner part with another b-bit string:

$$\mathbf{Pr}\left(\mathbf{Inner}^{\mathbf{lr}}_{\mathbf{a}}[i] \mid \neg\mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{ColKS}^{\mathbf{lr}}_{\mathbf{a}}[i]\right) \leq (\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] + \mathbf{1}_{\mathtt{C},\mathtt{KB}}[i])\,\frac{4(i-1)}{2^c}\,, \quad (29)$$

where we used that $\mathcal{M} + \mathcal{N} \leq 2^{b-1}$.

Case 2c. We then evaluate the event $\mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{a}}[i-1] \wedge \neg\mathbf{Inner}^{\mathbf{lr}}_{\mathbf{a}}[i] \wedge \neg\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]$. We will introduce the multicollision random variables Θ_{KBI} and Θ_{KBF} only during the sub-case that requires it, which is the reason why we did not switch to conditioned probabilities directly. If the i^{th} evaluation is from a permutation query or a leaky internal state evaluation, we upper bound the probability that the evaluation guesses correctly one of the following states:

- A state that is input of the leftmost key blinding: this case has already been handled with the bad event $\mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}}[i]$;
- A state that is output of the leftmost key blinding: the states in question are XORed with the keys before being leaked, and we can use multicollisions on the middle $c-k$ bits, since the key blinding evaluations are made in the forward direction. Let $\theta_{\mathrm{KBI}} \in \mathbb{N}$, then conditioned on $\Theta_{\mathrm{KBI}}[i] = \theta_{\mathrm{KBI}}$, this event is set with probability at most
$$\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{2\theta_{\mathrm{KBI}}}{2^k}\,;$$
- A state that is input of the rightmost key blinding: the same reasoning applies here. Let $\theta_{\mathrm{KBF}} \in \mathbb{N}$, then conditioned on $\Theta_{\mathrm{KBF}}[i] = \theta_{\mathrm{KBF}}$, this event is set with probability at most
$$\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{2\theta_{\mathrm{KBF}}}{2^k}\,;$$
- A challenge intermediate state, or output of the rightmost key blinding: the states in question are sampled uniformly and remain secret from the adversary. Therefore, this event is set with probability at most
$$\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i]\frac{4\left(\mathcal{M}_{E,C} - Q_{E,C} + Q_L\right)}{2^c}\,.$$

On the other hand, if the i^{th} evaluation is from an internal evaluation due to a challenge query, the event is set with a probability of at most

$$\mathbf{1}_{\mathtt{CH}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_{D,L}\right)}{2^c}\,.$$

If the i^{th} evaluation corresponds to a construction evaluation and is the output of the leftmost or rightmost permutation evaluation, then the output state is once again sampled in a permutation-consistent manner, and the event occurs with a probability of at most

$$\mathbf{1}_{\mathtt{C},\mathtt{KB}}[i]\frac{4\left(\mathcal{N} + \mathcal{M}_L - Q_{D,L}\right)}{2^c}\,.$$

We next consider the case when the i^{th} evaluation is from a construction evaluation, and the event is evaluated on the rightmost permutation input. Then, in the best case, the

adversary can choose among a certain set of states with full control on their outer parts, and then the key addition is applied on the middle k bits of the chosen state S. For all existing $(X, Y, \mathrm{d}) \in \tilde{\tau}[1 : i-1]$, the only candidates to set this event must have their outer part and rightmost $c-k$ bits set to those of S. Let $Z \in \{0,1\}^{b-k}$, and define C_Z as follows:

$$C_Z = \{(X, Y, \mathrm{d}) \in \tilde{\tau} \mid \lceil X \rceil_r \| \lfloor X \rfloor_{c-k} = Z\} \ .$$

C_Z counts the number of $(X, Y, \mathrm{d}) \in \tilde{\tau}$ such that X has its top r bits concatenated with its bottom $c-k$ bit fixed to Z. By the absence of $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i]$, the state S was obtained only from forward permutation evaluations, so that multicollisions can be used. Let $\theta_{\mathrm{KBF}} \in \mathbb{N}$. We define $\mathbf{1}_{\mathtt{C},\mathtt{KB_F},Z}[i \mid \theta_{\mathrm{KBF}}]$ to be the indicator function equal to 1 if and only if, conditioned on $\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}$, the evaluation number i is a rightmost key blinding, has it top r bits concatenated with its bottom $c-k$ bits equal to Z. Conditioned on $\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}$, this event is set with a probability of at most

$$\sum_{Z \in \{0,1\}^{b-k}} \mathbf{1}_{\mathtt{KB_f},Z}[i \mid \theta_{\mathrm{KBF}}] \frac{2C_Z}{2^k} \ .$$

Looking ahead, when summing over all queries and all possible values for multicollisions, we will have a term of the form:

$$\begin{aligned}
&\sum_i \sum_{\theta_{\mathrm{KBF}}} \sum_Z \mathbf{1}_{\mathtt{KB_f},Z}[i \mid \theta_{\mathrm{KBF}}] \frac{2C_Z}{2^k} \mathbf{Pr}\left(\Theta_{\mathrm{KBF}}[i] = \theta_{\mathrm{KBF}}\right) \\
&= \sum_Z \frac{2C_Z}{2^k} \sum_{\theta_{\mathrm{KBF}}} \mathbf{Pr}\left(\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}\right) \sum_i \mathbf{1}_{\mathtt{C},\mathtt{KB_F},Z}[i \mid \theta_{\mathrm{KBF}}] \\
&\leq \sum_Z \frac{2C_Z}{2^k} \sum_{\theta_{\mathrm{KBF}}} \mathbf{Pr}\left(\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}\right) \cdot \theta_{\mathrm{KBF}} \\
&\leq \frac{2(\mathcal{N}+\mathcal{M}) \cdot \mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k} \ ,
\end{aligned} \tag{30}$$

where we used (28). Therefore,

$$\begin{aligned}
&\mathbf{Pr}\left(\mathbf{GueS}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg \mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}[i-1] \wedge \neg \mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}}[i] \wedge \neg \mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}}[i]\right) \\
&\leq \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\left(\mathcal{M}_{E,C} - Q_{E,C} + Q_L\right)}{2^c} + \left(\mathbf{1}_{\mathtt{CH}}[i] + \mathbf{1}_{\mathtt{C},\mathtt{KBI}}[i]\right) \frac{2\left(\mathcal{N}+\mathcal{M}_L\right)}{2^c} \\
&+ \sum_{\theta_{\mathrm{KBF}} \in \mathbb{N}} \left(\mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\theta_{\mathrm{KBF}}}{2^k} + \sum_{Z \in \{0,1\}^{b-k}} \mathbf{1}_{\mathtt{C},\mathtt{KB_F},Z}[i \mid \theta_{\mathrm{KBF}}] \frac{2C_Z}{2^k} \right) \cdot \mathbf{Pr}\left(\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}\right) \\
&+ \sum_{\theta_{\mathrm{KBI}} \in \mathbb{N}} \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\theta_{\mathrm{KBI}}}{2^k} \mathbf{Pr}\left(\Theta_{\mathrm{KBI}} = \theta_{\mathrm{KBI}}\right) \\
&\leq \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\left(\mathcal{M}_{E,C} - Q_{E,C} + Q_L\right)}{2^c} + \left(\mathbf{1}_{\mathtt{CH}}[i] + \mathbf{1}_{\mathtt{C},\mathtt{KBI}}[i]\right) \frac{2\left(\mathcal{N}+\mathcal{M}_L\right)}{2^c} \\
&+ \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k} + \mathbf{1}_{\mathtt{P},\mathtt{CL}}[i] \frac{2\mathrm{mucol}\left(Q, 2^{c-k}\right)}{2^k} \\
&+ \sum_{\theta_{\mathrm{KBF}} \in \mathbb{N}} \mathbf{Pr}\left(\Theta_{\mathrm{KBF}} = \theta_{\mathrm{KBF}}\right) \sum_{Z \in \{0,1\}^{b-k}} \mathbf{1}_{\mathtt{C},\mathtt{KB_F},Z}[i \mid \theta_{\mathrm{KBF}}] \frac{2C_Z}{2^k} \ .
\end{aligned} \tag{31}$$

Combining (21) to (24), (26) and (29) to (31), we obtain

$$\begin{aligned}
\mathbf{Pr}\left(\mathbf{BAD2}_{\mathbf{a}}^{\mathbf{lr}}\right) \leq \ &\frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{2Q_D}{2^t} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} \\
&+ \frac{2\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M}) \cdot \mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k} \ .
\end{aligned} \tag{32}$$

Since both bounds (27) and (32) are valid upper bounds of the probability of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}$, we take the minimum of these two bounds and obtain

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{a}}\right) &\leq \min\left\{\mathbf{Pr}\left(\mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{a}}\right), \mathbf{Pr}\left(\mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{a}}\right)\right\} \\ &\leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{2Q_D}{2^t} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} \\ &+ \min\left\{\frac{14Q\left(\mathcal{N}+\mathcal{M}\right)}{2^k}, \frac{4\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M})\cdot \mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k}\right\}.\end{aligned}$$

We obtained an upper bound for the probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (20)), we conclude. □

6.4.2 Confidentiality

Let $\mathcal{A}$ be an adversary that makes at most $\mathcal{N}$ permutation queries, Q_E encryption queries of at most $\mathcal{M}_E$ blocks, and Q_D decryption queries of at most $\mathcal{M}_D$ blocks, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}^{\mu\text{-lr-conf}}_{\mathrm{Ascon},\mathcal{L}}(\mathcal{A})$, for any set of leakage functions that do not leak any information about the two outer permutation calls during the initialization and finalization phases. The adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \mathbf{Enc}^p_{K_m}, \left[\mathbf{Dec}^p_{K_m}\right]_L\right)^{\mu}_{m=1}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \$_m, \left[\mathbf{Dec}^p_{K_m}\right]_L\right)^{\mu}_{m=1}, p^{\pm}\right]$, where the adversary must be nonce-respecting with its queries to $\mathcal{O}_{2,m}$.

Similarly to the fact that the nonce-misuse authenticity proof can be used for bounding confidentiality in the nonce-misuse resilience setting, the proof of leakage resilience authenticity can be reused for leakage resilience confidentiality. We adopt the same terminology as in Section 6.4.1, with $\mathcal{O}_{1,m}$ and $\mathcal{O}_{3,m}$ will be referred to as *leaky oracles*, and $\mathcal{O}_{2,m}$ and $\mathcal{O}_{4,m}$ referred to as *challenge oracles*.

Setup. We adopt the same transcript and path notation as in Section 6.4.1, with the difference that there are no challenge decryption queries, which thus do not appear in the transcript, nor in the paths. We also generate mock intermediate states, following the same procedure as in the nonce-misuse resilience confidentiality proof in Section 6.2, and we can safely ignore leaky paths, since they do not have overlapping nonces. The extended transcript, released at the end of the interaction, can be derived as in Section 6.4.1. From $\tilde{\tau}$ we can reconstruct the two dictionaries $\mathtt{S}_{\mathrm{in}}$ and $\mathtt{S}_{\mathrm{out}}$, the set $\mathcal{P}$, the list of all permutation queries (X, Y, d), and the users' keys. Finally, we re-use the bad events defined in Section 6.4.1, except **Dec**, which does not apply there. In order to inherit the notation from this section, let us define $\mathcal{P}_{\mathrm{S}} = \emptyset$. The bad events are as follows:

$$\begin{aligned}\mathbf{ColK}^{\mathbf{lr}}_{\mathbf{c}} &: \mathbf{ColK}^{\mathbf{lr}}_{\mathbf{a}} \text{ (of Section 6.2)}; \qquad \mathbf{GueK}^{\mathbf{lr}}_{\mathbf{c}} : \mathbf{GueK}^{\mathbf{lr}}_{\mathbf{a}} \text{ (of Section 6.2)}; \\ \mathbf{ColS}^{\mathbf{lr}}_{\mathbf{c}} &: \mathbf{ColS}^{\mathbf{lr}}_{\mathbf{a}} \text{ (of Section 6.2)}; \qquad \mathbf{GueS}^{\mathbf{lr}}_{\mathbf{c}} : \mathbf{GueS}^{\mathbf{lr}}_{\mathbf{a}} \text{ (of Section 6.2)}; \\ \mathbf{Inner}^{\mathbf{lr}}_{\mathbf{c}} &: \mathbf{Inner}^{\mathbf{lr}}_{\mathbf{a}} \text{ (of Section 6.2)}; \\ \mathbf{BAD}^{\mathbf{lr}}_{\mathbf{c}} &: \mathbf{ColK}^{\mathbf{lr}}_{\mathbf{c}} \vee \mathbf{GueK}^{\mathbf{lr}}_{\mathbf{c}} \vee \mathbf{ColS}^{\mathbf{lr}}_{\mathbf{c}} \vee \mathbf{GueS}^{\mathbf{lr}}_{\mathbf{c}}; \\ \mathbf{BAD1}^{\mathbf{lr}}_{\mathbf{c}} &: \mathbf{BAD}^{\mathbf{lr}}_{\mathbf{c}}; \qquad \mathbf{BAD2}^{\mathbf{lr}}_{\mathbf{c}} : \mathbf{BAD}^{\mathbf{lr}}_{\mathbf{c}} \vee \mathbf{Inner}^{\mathbf{lr}}_{\mathbf{c}}.\end{aligned}$$

Probability of Good Transcripts. Since the bad events from Section 6.4.1 already handled collisions with challenge states from encryption queries, the same reasoning can be applied, so that the absence of $\mathbf{BAD}^{\mathbf{lr}}_{\mathbf{c}}$ guarantees that the intermediate states generated in the

ideal world are consistent with the structure of the mode Ascon-AE. Therefore, every transcript which is reachable in the real world is also reachable in the ideal world, and vice-versa, and for any transcript $\tilde{\tau}$ that does not set $\mathbf{BAD}_{\mathbf{c}}^{\mathbf{lr}}$, we have

$$\frac{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right)}{\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right)} \geq 1\,. \tag{33}$$

Probability of $\mathbf{BAD}_{\mathbf{c}}^{\mathbf{lr}}$ in the Ideal World. Compared to the evaluation done in Section 6.4.1, all challenge intermediate states are sampled at the end of the interaction, thus the evaluation of $\mathbf{BAD1}_{\mathbf{c}}^{\mathbf{lr}}$ (or $\mathbf{BAD2}_{\mathbf{c}}^{\mathbf{lr}}$) involving these challenge intermediate states is postponed at the end of the interaction. Overall, the same technique can be applied, and we obtain

$$\begin{aligned}\mathbf{Pr}\left(\mathbf{BAD}_{\mathbf{c}}^{\mathbf{lr}}\right) \leq\ & \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c} \\ & + \min\left\{\frac{14Q\left(\mathcal{N}+\mathcal{M}\right)}{2^k}, \frac{4\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M})\cdot \mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k}\right\}.\end{aligned}$$

We obtained an upper bound for probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (33)), we conclude. □

6.5 Proof of Theorem 5

Let $\mathcal{A}$ be an adversary that makes at most $\mathcal{N}$ permutation queries, Q_E encryption queries of at most $\mathcal{M}_E$ blocks, and Q_D decryption queries of at most $\mathcal{M}_D$ blocks, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon}}(\mathcal{A})$.

Let $\mathbf{BotDec}^p_{K_m}$ be the function that takes as input a tuple $(N, A, C, T) \in \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t$, computes internally $\mathbf{Dec}^p_{K_m}$, but always returns $\perp$. Therefore, the function $\left[\mathbf{BotDec}^p_{K_m}\right]_L$ always returns $\perp$, but leaks the internal states in the same way as $\left[\mathbf{Dec}^p_{K_m}\right]_L$. With this notation, the adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \left[\mathbf{Dec}^p_{K_m}\right]_L\right)_{m=1}^{\mu}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\left[\mathbf{Enc}^p_{K_m}\right]_L, \left[\mathbf{BotDec}^p_{K_m}\right]_L\right)_{m=1}^{\mu}, p^{\pm}\right]$. The following proof will be heavily based on the one of Section 6.4.1, the difference being that challenge queries are not present.

Setup. We adopt the same transcript and path notation as in Section 6.4.1, with the difference that there are no challenge queries. All intermediate states come from genuine permutation evaluations, so that no mock intermediate states are needed. However, we need to revise the definition of *superseded paths*. Recall that a decryption path $\mathtt{path} = (D_L, m, N, A, C, f)$ from a previous query is considered superseded by a later encryption path $\mathtt{path}' = (E_L, m, N, A, P, f)$ associated to an encryption query $(E_L, m, N, A', P', C', T) \in \tau$, if $|P| = |C|$ and $C = C'[1 : |C|]$. In the previous proofs, only decryption queries could be superseded by encryption queries, because intermediate states for decryption queries were generated at the end of the interaction. In contrast, now decryption queries can supersede encryption queries. In more detail, an encryption path (E_L, m, N, A, P, f) associated to an encryption query $(E_L, m, N, A', P', C, T)$ is superseded by a later decryption path $\mathtt{path}' = (D_L, m, N, A, C', f)$ if $|P| = |C'|$ and $C' = C[1 : |C'|]$. Let $\mathcal{P}_{\mathrm{S}}$ denote the set of paths that have been superseded according to this revised definition.

The extended transcript, released at the end of the interaction, can be derived as in Section 6.4.1. From $\tilde{\tau}$ we can reconstruct the two dictionaries $\mathtt{S}_{\text{in}}$ and $\mathtt{S}_{\text{out}}$, the set $\mathcal{P}$, the

list of all permutation queries (X, Y, d), and the users' keys. Finally, we re-use the bad events defined in Section 6.4.1 as follows:

$$\begin{aligned}
\mathbf{ColK}_{\mathbf{a}}^{\mathbf{sr}} &: \mathbf{ColK}_{\mathbf{a}}^{\mathbf{lr}} \text{ (of Section 6.4.1)}\,; \quad \mathbf{GueK}_{\mathbf{a}}^{\mathbf{sr}} : \mathbf{GueK}_{\mathbf{a}}^{\mathbf{lr}} \text{ (of Section 6.4.1)}\,;\\
\mathbf{GueS}_{\mathbf{a}}^{\mathbf{sr}} &: \mathbf{GueS}_{\mathbf{a}}^{\mathbf{lr}} \text{ (of Section 6.4.1)}\,; \quad \mathbf{Inner}_{\mathbf{a}}^{\mathbf{sr}} : \mathbf{Inner}_{\mathbf{a}}^{\mathbf{lr}} \text{ (of Section 6.4.1)}\,;\\
\mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}} &: \exists (D_L, m, N, A, C, T, \tilde{P}) \in \tilde{\tau} \text{ such that } \lfloor \mathtt{S}_{\mathrm{out}}[(D_L, m, N, A, C, 1)] \rfloor_t \oplus \lfloor K_m \rfloor_t = T, \text{ or}\\
&\quad \exists \mathtt{path} \in \mathcal{P} \text{ superseding } (D_L, m, N, A, C, 1) \text{ with } \lfloor \mathtt{S}_{\mathrm{out}}[\mathtt{path}] \rfloor_t \oplus \lfloor K_m \rfloor_t = T\,;\\
\mathbf{BAD}_{\mathbf{a}}^{\mathbf{sr}} &: \mathbf{ColK}_{\mathbf{a}}^{\mathbf{sr}} \vee \mathbf{GueK}_{\mathbf{a}}^{\mathbf{sr}} \vee \mathbf{GueS}_{\mathbf{a}}^{\mathbf{sr}} \vee \mathbf{Inner}_{\mathbf{a}}^{\mathbf{sr}} \vee \mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}}\,.
\end{aligned}$$

Because there are no more challenge paths, there is no bad event of the form **ColS**, and $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}}$ now includes leaky decryption queries. Moreover, now $\mathbf{Inner}_{\mathbf{a}}^{\mathbf{sr}}$ is incorporated in the main bad event.

Probability of Good Transcripts. As long as $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}}$ does not occur, the decryption queries in real world will always output $\perp$, so that for any good transcript $\tilde{\tau}$,

$$\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right) = \mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right). \tag{34}$$

Note that here, using the fundamental lemma of game playing [BR06] would have been sufficient, but we continue to use the H-coefficient technique for the sake of consistency.

Probability of $\mathbf{BAD}_{\mathbf{a}}^{\mathbf{sr}}$. The bounding of $\mathbf{BAD}_{\mathbf{a}}^{\mathbf{sr}}$ can be handled similarly to the second bounding in Section 6.4.1, with one key difference for the bad event $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}}$: this time, the adversary can submit forgeries with nonces associated to states that leaked. However, as long as $\neg\mathbf{GueS}_{\mathbf{a}}^{\mathbf{sr}}[i]$ holds, the adversary cannot predict the input to the key blinding call using its permutation queries. Moreover, as long a $\neg\mathbf{Inner}_{\mathbf{a}}^{\mathbf{sr}}[i]$ holds, every rightmost permutation evaluation made during a decryption query is unique, so that no two distinct decryption queries share the same input final state. Therefore, the bounding of the conditioned $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{sr}}[i]$ is the same, and we have

$$\begin{aligned}
\mathbf{Pr}\left(\mathbf{BAD}_{\mathbf{a}}^{\mathbf{sr}}\right) \leq\ & \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{N}+\mathcal{M}\right)}{2^k} + \frac{2Q_D}{2^t} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^c}\\
& + \frac{4\left(\mathcal{M}+\mathcal{N}\right)^2}{2^c} + \frac{6(\mathcal{N}+\mathcal{M})\cdot \mathrm{mucol}\left(\mathcal{N}+\mathcal{M}, 2^{c-k}\right)}{2^k}\,.
\end{aligned}$$

We obtained an upper bound for the probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (34)), we conclude. □

6.6 Proof of Theorem 6

Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q_E, \mathcal{M}_E, Q_D, \mathcal{M}_D)$, as in the theorem statement. Our goal is to upper bound $\mathbf{Adv}_{\mathrm{Ascon}}^{\mu\text{-rup-auth}}(\mathcal{A})$. Out of generosity, we assume that a query to $\mathbf{V}$ triggers first an evaluation to $\mathbf{D}$, and a query to $\mathbf{D}$ is followed by an evaluation of $\mathbf{V}$, and doing so only increases the adversary's success probability. Therefore, we can combine the oracles $\mathbf{D}$ and $\mathbf{V}$ back into a single oracle $\mathbf{DV}$, which first evaluates $\mathbf{D}$, then $\mathbf{V}$. Denote by $\mathbf{DBot}$ the oracle that computes $\mathbf{DV}$, releases the unverified plaintext, but instead of returning the output of $\mathbf{V}$, it always returns $\perp$.

With this notation, the adversary interacts either with the *real world* $\mathcal{W}_R$, which gives access to $\left[\left(\mathbf{Enc}_{K_m}^p, \mathbf{DV}_{K_m}^p\right)_{m=1}^{\mu}, p^{\pm}\right]$, or with the *ideal world* $\mathcal{W}_I$, which gives access to $\left[\left(\mathbf{Enc}_{K_m}^p, \mathbf{DBot}_{K_m}^p\right)_{m=1}^{\mu}, p^{\pm}\right]$. Here, the adversarial complexity is $(Q_D, \mathcal{M}_D)$ for the

oracle $\mathcal{O}_{2,m}$. By abuse of notation, we will continue to refer to the oracle $\mathcal{O}_{2,m}$ as a decryption oracle.

The following proof builds on the nonce-misuse authenticity proof from Section 6.2. The key difference lies in how decryption queries are handled: in the current case, all states are computed on-the-fly rather than at the end of the interaction. We will make adjustments similar to those made when adapting the leakage resilience proof (in Section 6.4.1) for the state-recovery proof (in Section 6.5).

Setup. The transcript notation now needs to account for the unverified plaintext released by the decryption oracle. It is defined as follows:

- A forward (resp., inverse) permutation query with input X and output Y generates the transcript element (X, Y, fwd) (resp., (X, Y, inv));
- An encryption query with user m, input (N, A, P), and output (C, T) generates the transcript element (E, m, N, A, P, C, T);
- A decryption query with user m, input (N, A, C, T), output unverified plaintext P, and verification output V generates the transcript element (D, m, N, A, C, P, T, V).

We adopt the same path notation as in Section 6.2, but adapt the notion of *superseded paths* as done in Section 6.5. Let $\mathcal{P}_{\mathrm{S}}$ denote the set of paths that have been superseded. All intermediate states come from genuine permutation evaluations, so that no mock intermediate states are needed. The extended transcript, released at the end of the interaction, can be derived as in Section 6.2. Finally, we re-use the bad events defined in Section 6.2 as follows:

$$\begin{aligned}
\mathbf{ColK}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{ColK}_{\mathbf{a}}^{\mathbf{m}} \text{ (of Section 6.2)}; \quad & \mathbf{GueK}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{GueK}_{\mathbf{a}}^{\mathbf{m}} \text{ (of Section 6.2)}; \\
\mathbf{ColS}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{ColS}_{\mathbf{a}}^{\mathbf{m}} \text{ (of Section 6.2)}; \quad & \mathbf{GueS}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{GueS}_{\mathbf{a}}^{\mathbf{m}} \text{ (of Section 6.2)}; \\
\mathbf{Dec}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{Dec}_{\mathbf{a}}^{\mathbf{m}} \text{ (of Section 6.2)}; & & \\
\mathbf{BAD}_{\mathbf{a}}^{\mathbf{rup}} &: \mathbf{ColK}_{\mathbf{a}}^{\mathbf{rup}} \vee \mathbf{GueK}_{\mathbf{a}}^{\mathbf{rup}} \vee \mathbf{ColS}_{\mathbf{a}}^{\mathbf{rup}} \vee \mathbf{GueS}_{\mathbf{a}}^{\mathbf{rup}} \vee \mathbf{Dec}_{\mathbf{a}}^{\mathbf{rup}}.
\end{aligned}$$

Probability of Good Transcripts. Similar to the state-recovery proof of Section 6.5, as long as $\mathbf{Dec}_{\mathbf{a}}^{\mathbf{rup}}$ does not occur, the decryption queries in real world will always output $\perp$, so that for any good transcript $\tilde{\tau}$,

$$\mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_R\right] \text{ generates } \tilde{\tau}\right) = \mathbf{Pr}\left(\mathcal{A}\left[\mathcal{W}_I\right] \text{ generates } \tilde{\tau}\right). \tag{35}$$

Again, using the fundamental lemma of game playing [BR06] would have been sufficient, but we continue to use the H-coefficient technique for the sake of consistency.

Probability of BAD in the Ideal World. The same bounding technique as in Section 6.2 can be done. Indeed, the bad events defined previously are general enough to treat encryption and decryption states the same way, focusing only on equality on the inner part of the states. In this case, there are no more events that can be set at the end of the interaction. Therefore,

$$\mathbf{Pr}\left(\mathbf{BAD}_{\mathbf{a}}^{\mathbf{rup}}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{2\mu\left(\mathcal{M}+\mathcal{N}\right)}{2^{k}} + \frac{18\mathcal{M}\left(\mathcal{M}+\mathcal{N}\right)}{2^{c}} + \frac{2Q_D}{2^{t}}.$$

We obtained an upper bound for the probability that the ideal world generates a bad transcript. Using the H-coefficient technique, and the fact that the ratio of good transcript is lower-bounded by one (cf., (35)), we conclude. □

7 Generic Attacks for Ascon-AE

We include the attacks of Propositions 1, 3–5, and 7 here, in Sections 7.1–7.5, respectively.

7.1 Proof of Proposition 1

In each of the following sections, we describe the adversaries as stated in the proposition. These are all forgery attacks, but they can be easily converted into distinguishing attacks by returning "ideal" when the attack fails, and "real" otherwise (see for instance the last step in Section 7.3).

7.1.1 Adversary $\mathcal{A}_1$

Recall that $\mathcal{A}_1$ has resources satisfying $Q_D \approx 2^t$. A simple matching attack consists of submitting $\approx 2^t$ arbitrary decryption queries, each with a different tag.

7.1.2 Adversary $\mathcal{A}_2$

Recall that $\mathcal{A}_2$ has resources satisfying $\mathcal{N} \approx 2^k/\mu$. This term corresponds to the probability that the adversary guesses one of the user keys. Consider the following attack:

1. Let $N \in \{0,1\}^n$. For $m \in [\![1, \mu]\!]$ make an encryption query, with user m, with input $(N, \emptyset, 0^r)$, and denote the ciphertext by C^m and the tag by T^m;
2. For $i \in [\![1, \mathcal{N}]\!]$, sample $L_i \xleftarrow{\$} \{0,1\}^k$, and compute $\mathbf{Enc}^p_{L_i}(N, \emptyset, 0^r)$, get a ciphertext $\tilde{C}^i$;
3. With probability $\approx \frac{\mu\mathcal{N}}{2^k} \approx 1$, there exists a L_i from step 2 that collides with a key K_m of a user m. The adversary can check this by observing that $\tilde{C}^i = C^m$. In the following, we assume that this is the case;
4. Let $P \in \{0,1\}^*$, and $N' \neq N$. Compute $(C, T) = \mathbf{Enc}^p_{L_i}(N', \emptyset, P)$;
5. Submit a forgery for user m with input $(N', \emptyset, C, T)$.

In order to reduce the probability of a false positive in step 3 to a negligible probability, the encryption queries in step 1 can be extended with $\gamma = \mathcal{O}\left(\lceil \frac{b}{r} \rceil\right)$ plaintext blocks.

7.1.3 Adversary $\mathcal{A}_3$

Recall that $\mathcal{A}_3$ has resources satisfying $(\mathcal{M}_E + 2^r)\mathcal{N} \approx 2^b$. At a high level, the following attack involves guessing an intermediate state generated during encryption queries and using it to create a forgery. Let $\gamma \in \mathbb{N}$ be a parameter. As in the previous attack, γ serves to mitigate the probability to obtain false positives, and we suggest taking $\gamma = \mathcal{O}\left(\lceil \frac{b}{r} \rceil\right)$. For simplicity of presentation, we will assume that $\mu \cdot 2^n \geq \mathcal{M}_E$, but the attack can be easily adapted by stretching the block length of the encryption queries by a logarithmic factor. Let $\left(m^i, N^i\right)_i$ be a family of $\mathcal{M}_E$ pairwise distinct user indices and nonces. The attack operates a follows:

1. **Encryption queries:** for $i = 1, \ldots, \mathcal{M}_E$, make an encryption query with user m^i with input $\left(N^i, \emptyset, 0^{r(\gamma+1)}\right)$, and denote the ciphertexts by $C^i_0, C^i_1, \ldots, C^i_\gamma$.

   This step requires $\mathcal{O}(\mathcal{M}_E)$ encryption queries with $\mathcal{O}(\gamma)$ blocks each;

2. **State guessing:**

   (a) Denote by $\nu \in \{0,1\}^r$ the outer part that occurs most frequently among $(C^i_0)_{i \in [\![1, \mathcal{M}_E]\!]}$;

(b) For $l \in [\![1, \mathcal{N}]\!]$, sample $X^l \xleftarrow{\$} \{0,1\}^c$, and compute $p^f(\nu \| X^l)$ for $f \in [\![1, \gamma]\!]$;

(c) If there exists some $l \in [\![1, \mathcal{N}]\!]$, $i \in [\![1, \mathcal{M}_E]\!]$ such that for all $f \in [\![0, \gamma]\!]$, $\lceil p^f(\nu \| X^l) \rceil_r = C_f^i$, then we consider that the adversary has successfully guessed the intermediate state with user m^i, nonce N^i, empty associated data, and after having absorbed the plaintext block 0^r;

(d) Let $S_{\text{in}} := \nu \| X^l$, S_{fin} be the state of the associated encryption query right before the last key blinding, T be the tag of the associated encryption query, $N := N^i$, and $m := m^i$.

This step requires $\gamma\mathcal{N}$ permutation queries. As for the success probability, note that γ allows to reduce the influence of false positives. Therefore, the success probability is almost the probability that the adversary permutation query history contains a successful guess, which is itself $\approx \frac{\mathcal{M}_E \mathcal{N}}{2^b} + \frac{\mathcal{N}}{2^c} \approx 1$;

3. **State binding:** The goal of this step is to connect S_{in} and S_{fin} by a sequence of message blocks different from $(0^{r(\gamma+1)}, 1 \| 0^{r-1})$. To achieve this, we use an attack introduced by the designers of the sponge [BDPV07], which relies on finding inner collisions. Let us first for simplicity assume that $r > c/2$. It consists of the following steps:

   (a) Compute $X^i := p\left(S_{\text{in}} \oplus P_1^i \| 0^c\right)$ for $\mathcal{N}$ different P_1^is, distinct from 0^r;[9]

   (b) Compute $Y^i := p^{-1}\left(S_{\text{fin}} \oplus P_3^i \| 0^c\right)$ for $\mathcal{N}$ different P_3^is, distinct from 0^r;

   (c) If there exists $i, j \in [\![1, \mathcal{N}]\!]$ such that $\lfloor X_i \rfloor_c = \lfloor Y_j \rfloor_c$, let $P_2 := \lceil X_i \rceil_r \oplus \lceil Y_j \rceil_r$, and output (P_1^i, P_2, P_1^j); else abort.

   Note that the returned sequence of message blocks corresponds to a valid padding. In the setting where $\mathcal{N} \geq 2^r$ and $r \leq c/2$, the attack can be extended by making multiple sequential absorb calls in steps 3a and 3b. In the following, we denote the output sequence of message blocks by $(P_1, \ldots, P_d)$.

   This step requires $2\mathcal{N}$ permutation queries. The step succeeds with probability $\approx \min\left\{1, \frac{\mathcal{N}^2}{2^c}\right\}$. Given that $(\mathcal{M}_E + 2^r)\mathcal{N} \approx 2^b$ and $\mathcal{M}_E \ll \mathcal{N}$, this gives a success probability of ≈ 1;

4. **Compute the ciphertexts:** Initialize $S \leftarrow S_{\text{in}}$. Then, for $f \in [\![1, d]\!]$, compute $\tilde{C}_f \leftarrow \lceil S \rceil_r \oplus P_f$, and update $S \leftarrow p(S \oplus (P_f \| 0^c))$;

5. **Forgery:** Let $l = |\text{unpad}_r(P_1 \| \cdots \| P_d)|$. Submit a forgery with input $(m, N, \emptyset, (\tilde{C}_1 \| \cdots \| \tilde{C}_d)[0:l], T)$.

Overall, the attack requires $\mathcal{O}(\mathcal{M}_E)$ encryption queries, one decryption query, and $\mathcal{O}(\mathcal{N})$ permutation queries, and succeeds with high probability. □

7.2 Proof of Proposition 3

Recall that $\mathcal{A}$ has resources satisfying $\mathcal{M}_E \mathcal{N} \approx 2^c$. The attack is similar to the one described in Section 7.1.3, with the main difference being that in this case, the adversary makes use of the nonce-misuse setting to set the outer parts of the states to a value of its choice. Let $\gamma \in \mathbb{N}$ be a parameter. As in the previous attacks, γ serves to mitigate the probability to obtain false positives, and we suggest taking $\gamma = \mathcal{O}\left(\lceil \frac{b}{r} \rceil\right)$. For simplicity, we describe the attack in the case where $\mu \cdot 2^n \geq \mathcal{M}_E$, similar to the nonce-respecting case. Let $\left(m^i, N^i\right)_i$ be a family of $\mathcal{M}_E$ pairwise distinct user indices and nonces. The attack operates a follows:

[9] To generalize the attack to any input sequence of message blocks $(B_1, \ldots, B_l)$, the P_1^is can be chosen different from B_1.

1. **Encryption queries:** For $i = 1, \dots \mathcal{M}_E$, do the following:

   (a) Make an encryption query with user m^i with input $\left(N^i, \emptyset, 0^r\right)$, and denote by C^i the obtained ciphertext;

   (b) Make an encryption query with user m^i with input $\left(N^i, \emptyset, C^i \| 0^{\gamma r}\right)$, get ciphertexts $\left(C_0^i, C_1^i, \dots, C_\gamma^i\right)$, and tag T^i.

   Note that in the second permutation evaluation made in the context of the query from step (b), the outer part of the state is set to 0^r, so that necessarily $C_0^i = 0^r$. This step requires $\mathcal{O}\left(\mathcal{M}_E\right)$ encryption queries, each with $\mathcal{O}\left(\gamma\right)$ blocks, and each nonce is re-used twice;

2. **State guessing:** This step is identical to step 2 of Section 7.1.3, except that in this case, the permutation queries of the adversary now have all their outer r bits fixed to 0^r. Thus, the adversary only needs to guess one of the rightmost c bits of the intermediate states obtained during encryption queries. This speeds up the success probability to $\approx \frac{\mathcal{M}_E \mathcal{N}}{2^c} \approx 1$;

3. **State binding:** Same as step 3 of Section 7.1.3;

4. **Compute the ciphertexts:** Same as step 4 of Section 7.1.3;

5. **Forgery:** Same as step 5 of Section 7.1.3.

Overall, the attack requires $\mathcal{O}\left(\mathcal{M}_E\right)$ encryption queries where each nonce is repeated twice, one decryption query, and $\mathcal{O}\left(\mathcal{N}\right)$ permutation evaluations. The state guessing attack (step 2) succeeds with probability $\approx \frac{\mathcal{M}_E \mathcal{N}}{2^c} \approx 1$, the state binding attack (step 2) succeeds with probability $\approx \min\left\{1, \frac{\mathcal{N}^2}{2^c}\right\} \approx 1$, so that this attack succeeds with probability close to 1. □

7.3 Proof of Proposition 4

We give below a key recovery attack that exploits nonce-misuse encryption queries. Recall that $\mathcal{A}^{\text{conf}}$ and $\mathcal{A}^{\text{auth}}$ have resources satisfying $\mathcal{M}_E \mathcal{N} \approx 2^c$. The strategies of $\mathcal{A}^{\text{conf}}$ and $\mathcal{A}^{\text{auth}}$ are identical, except in the final step. In the phases of the attack shared with $\mathcal{A}^{\text{auth}}$, $\mathcal{A}^{\text{conf}}$ makes encryption queries only to the non-challenge oracle (i.e., $\mathcal{O}_{1,m}$), allowing thus nonce reuse. The attack operates as follows:

1. **Encryption queries and state guessing:** Apply steps 1 and 2 from the attack described in Section 7.2. If the attack succeeds, denote by S_{in} the state guessed and let $S_{\text{out}} = p\left(S_{\text{in}}\right)$. Let m, N, and P be the user, nonce, and plaintext block sequence associated to this state (i.e., after absorbing P, one gets the permutation input S_{in}), respectively.

   This step requires $\mathcal{O}\left(\mathcal{M}_E\right)$ nonce-misuse encryption queries with $\mathcal{O}\left(1\right)$ blocks each, where each nonce is repeated twice, and $\mathcal{O}\left(\mathcal{N}\right)$ permutation queries. The step succeeds with probability $\approx \frac{\mathcal{M}_E \mathcal{N}}{2^c} \approx 1$;

2. **State expansion:** For $i \in [\![1, \mathcal{M}_E]\!]$, make a forward permutation query $p(S_{\text{out}} \oplus (P_2^i \| 0^c))$.[10] We obtain a family of states along with their paths:

$$S^i = 0^r \| \lfloor p(S_{\text{out}} \oplus (P_2^i \| 0^c)) \rfloor_c \,,$$
$$\texttt{path}^i = \left(m, N, \emptyset, (P, P_2^i, \lceil p(S_{\text{out}} \oplus (P_2^i \| 0^c)) \rceil_r)\right) \,.$$

[10] If $2^r < \mathcal{M}_E$, this step can be extended by making cascaded permutation calls.

Note that all states S^is have their outer r bits fixed to 0^r. Some of the states S^i will later be interpreted as states right before the last key blinding.

This step requires $\mathcal{M}_E$ permutation queries;

3. **Filtering:** Let mucol $\in \{0,1\}^{c-k}$ be such that the size of the following set is maximal:

$$I_{\text{mucol}} = \left\{ i \in [\![1, \mathcal{M}_E]\!] \mid \lfloor S^i \rfloor_{c-k} = \text{mucol} \right\} .$$

In other words, the states S^i are filtered by selecting those with the rightmost $c-k$ bits to the value that maximizes the number of S^is. We expect that $|I_{\text{mucol}}| \geq \max\left\{1, \frac{\mathcal{M}_E}{2^{c-k}}\right\}$ with high probability. Looking ahead, in the regime where $\frac{\mathcal{M}_E}{2^{c-k}} \leq 1$, then the attack targeting the term $\frac{\mu\mathcal{N}}{2^k}$ from Section 7.1.2 is better. Therefore, we assume $\frac{\mathcal{M}_E}{2^{c-k}} > 1$ in the following;

4. **Construction queries:** For each $i \in I_{\text{mucol}}$, make an encryption query with input $\texttt{path}^i$, and get tag T^i.

This step requires on expectation $\frac{\mathcal{M}_E}{2^{c-k}}$ encryption queries, all using the same nonce;

5. **Key guessing:** From the previous step, the adversary has made $\approx \frac{\mathcal{M}_E}{2^{c-k}}$ encryption queries, where the inputs to their last permutation query (during the key blinding) have $b-k$ bits set to a value known by the adversary. These bits are always in fixed positions, i.e., the leftmost r bits and the rightmost $c-k$ bits. The following step consists of trying to guess the remaining k bits of those states.

For $j = 1, \ldots, \mathcal{N}$, do the following:

   (a) Sample $X^j \xleftarrow{\$} \{0,1\}^k$, and make the permutation query $p\left(0^r \| X^j \| \text{mucol}\right)$;

   (b) For each $i \in I_{\text{mucol}}$:

      i. Compute $K^{ij} = \lceil \lfloor S^i \rfloor_c \rceil_k \oplus X^j$;

      ii. If $\left\lfloor \lfloor p\left(0^r \| X^j \| \text{mucol}\right) \rfloor_k \oplus K^{ij} \right\rfloor_t = T^i$, then K^{ij} is a key candidate;

      iii. For each key candidate, the adversary checks whether the obtained ciphertexts and the state S^i from the encryption query number i match those obtained with a direct evaluation of $\mathbf{Enc}^p_{K^{ij}}$. This step allows to determine whether K^{ij} is a false positive or a correct key.

Checking one false positive in step 5(b)iii costs $\mathcal{O}(1)$ permutation queries. In order to not blow up the permutation complexity of the adversary, we allow a small number of false positives per permutation query, hence the requirement $\frac{\mathcal{M}_E}{2^{c-k+t}} \leq 1$, or equivalently $\mathcal{N} \geq 2^{k-t}$, as stated in the proposition.

The probability that the adversary, via one of the permutation calls made in this step, guesses the k bits of one of the $i \in I_{\text{mucol}}$ encryption queries is

$$\approx \frac{\mathcal{N}\,|I_{\text{mucol}}|}{2^k} \approx \frac{\mathcal{M}_E \mathcal{N}}{2^c} \approx 1 ;$$

6. **Last step:** Let K be the guessed key. Depending on whether the adversary is against confidentiality or authenticity, do the following:

   - $\mathcal{A}^{\text{conf}}$: make an evaluation with $\mathbf{Enc}^p_K$ with a new nonce, obtain several ciphertext blocks (e.g., take $\gamma = \mathcal{O}\left(\lceil \frac{b}{r} \rceil\right)$), and make the same query to the challenge oracle. If the ciphertext blocks coincide with the challenge oracle, return 1, else return 0. The distinguishing advantage of the adversary is close to 1;

- $\mathcal{A}^{\text{auth}}$: make an evaluation with $\mathbf{Enc}_K^p$ with a new nonce, obtain a ciphertext and tag, and submit it to the challenge decryption oracle.

Overall, the attack requires $\mathcal{O}(\mathcal{M}_E)$ encryption queries and $\mathcal{O}(\mathcal{N})$ permutation queries, and succeeds with probability close to 1. □

7.4 Proof of Proposition 5

The attack shares similarities with the nonce-misuse resilience key-recovery attack presented in Section 7.3, with the key difference being that the adversary has direct access to the intermediate states, and thus does not need to do step 1 of this attack. This gives more freedom in the parametrization of the attack. The strategies of $\mathcal{A}^{\text{conf}}$ and $\mathcal{A}^{\text{auth}}$ are identical, except in the final step. In the phases of the attack shared with $\mathcal{A}^{\text{auth}}$, $\mathcal{A}^{\text{conf}}$ makes encryption queries only to the non-challenge oracle (i.e., $\mathcal{O}_{1,m}$), thus leaky encryption queries. Here, the attack is parametrized by the maximum number of allowed encryption queries, thus allowing for a tradeoff between offline and online complexity. We make a case distinction depending on the chosen parameter regime.

Case 1, when $\max\{2^{c/2}, 2^k/Q_E\} = 2^{c/2}$. In this situation, the number of allowed encryption queries is not a limiting factor. In that case, we have $\mathcal{N} \approx 2^{c/2}$, and the attack operates as follows:

1. **State leaking:** Let $m \in \{0,1\}^\mu$, $N \in \{0,1\}^n$, $\text{PT} \in \{0,1\}^*$, and $P \leftarrow \text{pad}_r^{10^*}(\text{PT})$. Make an encryption query to the leaky oracle with user m and input $(N, \emptyset, \text{PT})$, and obtain S_{in}, the state after having absorbed P. Let $S_{\text{out}} = p(S_{\text{in}})$;

2. **State expansion:** For $i \in [\![1, \mathcal{N}]\!]$, make a forward permutation query $p(S_{\text{out}} \oplus (P_2^i \| 0^c))$.[11] We obtain a family of states along with their paths:

$$S^i = 0^r \| \lfloor p(S_{\text{out}} \oplus (P_2^i \| 0^c)) \rfloor_c ,$$
$$\texttt{path}^i = \left(m, N, \emptyset, (P, P_2^i, \lceil p(S_{\text{out}} \oplus (P_2^i \| 0^c)) \rceil_r)\right) .$$

   We will interpret S^i as a state right before the last key blinding.

   This step requires $\mathcal{N}$ permutation queries;

3. **Filtering:** Let $\text{mucol} \in \{0,1\}^{c-k}$ be such that the size of the following set is maximal:

$$I_{\text{mucol}} = \left\{ i \in [\![1, \mathcal{N}]\!] \mid \lfloor S^i \rfloor_{c-k} = \text{mucol} \right\} .$$

   We expect that $|I_{\text{mucol}}| \geq \frac{\mathcal{N}}{2^{c-k}}$ with high probability;

4. **Construction queries:** For each $i \in I_{\text{mucol}}$, make an encryption query with input $\texttt{path}^i$, and get tag T^i.

   This step requires on expectation $\frac{\mathcal{N}}{2^{c-k}} \approx 2^{k-c/2} \leq Q_E$ encryption queries;

5. **Key guessing:** Same as step 5 of Section 7.3.

   Since the size of I_{mucol} is different, the cost computation is a bit different as well. The number of false positives per permutation query is equal to

$$\frac{|I_{\text{mucol}}|}{2^t} = \frac{\mathcal{N}}{2^{c-k+t}} .$$

[11] If $2^r < \mathcal{N}$, this step can be extended by making cascaded permutation calls.

Similarly to the prior attack, we limit the number of false positives per permutation query to at most a small constant, hence the requirement $\mathcal{N} \geq 2^{k-t}$, as stated in the proposition.

The probability that the adversary, via one of the permutation calls made in this step, guesses the k bits of one of the $i \in I_{\text{mucol}}$ encryption queries is

$$\approx \frac{\mathcal{N}\,|I_{\text{mucol}}|}{2^k} \approx \frac{\mathcal{N}^2}{2^c} \approx 1\,;$$

6. **Last step:** Same as step 6 from Section 7.3.

Overall, this attack requires $\approx 2^{k-c/2}$ encryption queries and $\mathcal{O}(\mathcal{N})$ permutation queries, and succeeds with probability close to 1.

Case 2, when $\max\{2^{c/2}, 2^k/Q_E\} = 2^k/Q_E$. In this situation, the number of allowed encryption queries is a limiting factor, so that the offline complexity needs to compensate for that, so that $\mathcal{N} \approx 2^k/Q_E$. The performed steps are the same as in case 1, but the parametrization is different. The attack operates as follows:

1. **State leaking:** Same as step 1 of case 1;

2. **State expansion:** Same as step 2 of case 1, with the difference that the number of permutation calls made here is $Q_E 2^{c-k}$. Note that, since $\mathcal{N} \approx 2^k/Q_E$, this step costs $2^c/\mathcal{N} \leq 2^{c/2} \leq \mathcal{N}$ permutation queries;

3. **Filtering:** Same as step 3 of case 1. This time, we expect that $|I_{\text{mucol}}| \geq Q_E$ with high probability;

4. **Construction queries:** Same as step 4 of case 1. This time, the number of encryption queries is Q_E;

5. **Key guessing:** Same as step 5 of case 1. This time, the number of false positives per permutation query is equal to

$$\frac{|I_{\text{mucol}}|}{2^t} = \frac{Q_E}{2^t}\,.$$

Again, the constraint $\mathcal{N} \geq 2^{k-t}$ guarantees that every permutation query has no more than a small number of false positives.

The probability that the adversary, via one of the permutation calls made in this step, guesses the k bits of one of the $i \in I_{\text{mucol}}$ encryption queries is

$$\approx \frac{\mathcal{N}\,|I_{\text{mucol}}|}{2^k} \approx \frac{Q_E \mathcal{N}}{2^k} \approx 1\,.$$

6. **Last step:** Same as step 6 of case 1.

Overall, this attack requires Q_E encryption queries and $\mathcal{O}(\mathcal{N})$ permutation queries, and succeeds with probability close to 1.

With the case distinction performed depending on whether the number of allowed encryption queries is limiting or not, we obtained an attack with complexities as stated in the proposition. □

7.5 Proof of Proposition 7

At a high level, this is the forgery attack of Section 7.1.3, but with the state guessing step removed, as the leaky oracles give access to the states.

1. **State leaking:** Let $m \in \{0,1\}^{\mu}$, $N \in \{0,1\}^{n}$, $\mathrm{PT} \in \{0,1\}^{*}$, and $P \leftarrow \mathrm{pad}_{r}^{10^{*}}(\mathrm{PT})$. Make an encryption oracle to the leaky oracle with user m and input $(N, \emptyset, \mathrm{PT})$, and obtain S, the state right after the first key blinding. Let S_{fin} be the state after having absorbed P (i.e., right before the last key blinding);
2. **State binding:** Same as step 3 of Section 7.1.3 with states S and S_{fin}, and plaintext block P;
3. **Compute the ciphertexts:** Same as step 4 of Section 7.1.3;
4. **Forgery:** Same as step 5 of Section 7.1.3.

Overall, the attack requires one encryption query, one decryption query, and $\mathcal{O}(\mathcal{N})$ permutation queries. The state binding attack (step 2) succeeds with probability $\approx \frac{\mathcal{N}^2}{2^c} \approx 1$. □

8 Ascon-Hash/Ascon-(C)XOF Modes and Their Security

We describe the mode underlying Ascon-Hash/Ascon-(C)XOF in Section 8.1, the security models in Section 8.2, and the security of the constructions in Section 8.3.

8.1 Description of the Modes

Ascon-Hash, Ascon-XOF, as well as Ascon-CXOF are based on the sponge construction [BDPV07]. Let $b, c, r \in \mathbb{N}$ such that $b = r + c$, and let p be a cryptographic permutation over b bits. The sponge construction takes as input a plaintext $P \in \{0,1\}^{*}$, and a length $l \in \mathbb{N}$. It returns a digest $Z \in \{0,1\}^{*}$ with $|Z| = l$. Then, Ascon-XOF^{p} is defined as follows:

$$\begin{aligned} \text{Ascon-XOF}^{p} : \{0,1\}^{*} \times \mathbb{N} &\longrightarrow \{0,1\}^{*} \\ (P, l) &\longrightarrow Z \in \{0,1\}^{l}\,. \end{aligned}$$

The construction is illustrated in Figure 4. Here, $IV \in \{0,1\}^{b}$ is a fixed initialization value. The Ascon specification [DEMS19, SMKK24] also specifies Ascon-Hash and Ascon-CXOF, the main difference being in the choice of initialization value IV. Ascon-CXOF then further distinguishes itself from the others in the fact that it additionally takes a customization string $C \in \{0,1\}^{\leq 2048}$, and incorporates it into the padded message with a length encoding at the beginning. The Ascon draft standard [SMKK24] specifies Ascon-Hash, Ascon-XOF, and Ascon-CXOF to operate with capacity $c = 256$ and rate $r = 64$. The digest size of Ascon-Hash is 256 whereas for Ascon-XOF and Ascon-CXOF it is unlimited.

8.2 Security Model

The main security model for hashing is indifferentiability, of Maurer et al. [MRH04]. This model was tailored to cryptographic hash functions by Coron et al. [CDMP05], though that model also applies to XOFs. Intuitively, the Ascon-XOF XOF based on random permutation p is indifferentiable from a random oracle \$ if there exists a simulator **Sim** with oracle access to the random oracle \$ such that (Ascon-XOF^{p}, $p^{\pm}$) is indistinguishable from (\$, **Sim**[\$]$^{\pm}$).

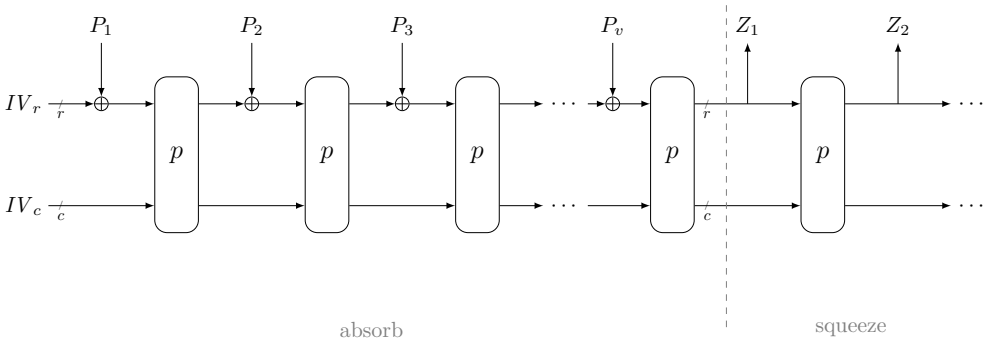

Figure 4: The Ascon-XOF mode of operation. Here, P is injectively padded as $(P_1, \ldots, P_v) \leftarrow \mathrm{pad}_r^{10^*}(P)$. Here, for graphical convenience, the IV is split as $IV = (IV_r, IV_c)$, with $IV_r \in \{0,1\}^r$, $IV_c \in \{0,1\}^c$.

Definition 9. Consider the Ascon-XOF mode of Section 8.1. Let \$ be a random function, $p \xleftarrow{\$} \mathrm{Perm}(b)$, and $\mathbf{Sim}[\$]^{\pm}$ be a two-sided algorithm with oracle access to \$. The indifferentiability of Ascon-XOF with respect to simulator **Sim** against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{indiff}}_{\text{Ascon-XOF},\mathbf{Sim}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\text{Ascon-XOF}^p, p^{\pm}\,;\,\$, \mathbf{Sim}[\$]^{\pm}\right).$$

In indifferentiability, the adversarial resources are counted in the number of accumulated permutation evaluations $\mathcal{N}$ that would be made in the left world.

However, in certain applications, it suffices to focus on the classical notions of collision, preimage, and second preimage security. We will consider these notions in the random permutation model, where we take collision resistance, (everywhere) preimage resistance, and (everywhere) second preimage of Rogaway and Shrimpton [RS04]. Here, we require that the *minimal* output size is fixed to some value ν. The second preimage security is indexed by a parameter κ that specifies the maximal length of the preimage.

Definition 10. Consider the Ascon-XOF mode of Section 8.1, and let $p \xleftarrow{\$} \mathrm{Perm}(b)$. Let $\kappa, \nu \in \mathbb{N}$.

- The collision resistance of Ascon-XOF against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{col}[\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}\left[p^{\pm}\right] \to (P, P') \;\begin{array}{l}\text{such that } P \neq P' \text{ and} \\ \text{Ascon-XOF}^p(P,\nu) = \text{Ascon-XOF}^p(P',\nu)\end{array}\right);$$

- The (everywhere) preimage resistance of Ascon-XOF against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{pre}[\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) = \max_{Z \in \{0,1\}^{\nu}} \mathbf{Pr}\left(\mathcal{A}\left[p^{\pm}\right](Z) \to P \;\text{ such that } \text{Ascon-XOF}^p(P,\nu) = Z\right);$$

- The (everywhere) second preimage resistance of Ascon-XOF against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{sec}[\kappa,\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) = \max_{P \in \{0,1\}^{\leq \kappa}} \mathbf{Pr}\left(\mathcal{A}\left[p^{\pm}\right](P) \to P' \;\begin{array}{l}\text{such that } P \neq P' \text{ and} \\ \text{Ascon-XOF}^p(P,\nu) = \text{Ascon-XOF}^p(P',\nu)\end{array}\right).$$

Andreeva et al. [AMP10, Appendix A] gave the reasoning why indifferentiability implies collision, preimage, and second preimage resistance. In detail, we have the following reduction.

Lemma 3 ([AMP10]). *Consider the* Ascon-XOF *mode of Section 8.1. Let* \$ *be a random function,* $p \xleftarrow{\$} \mathrm{Perm}(b)$*, and* $\mathbf{Sim}[\$]^{\pm}$ *be any two-sided algorithm with oracle access to* \$*. Let* $\kappa, \nu \in \mathbb{N}$*. Let* $\mathrm{x} \in \{\mathrm{col}[\nu], \mathrm{pre}[\nu], \mathrm{sec}[\kappa, \nu]\}$*, and let* $\mathcal{A}$ *be an* x *adversary with complexity* $\mathcal{N}$*. There exists an indifferentiability adversary* $\mathcal{A}'$ *with respect to simulator* **Sim** *with complexity* $\mathcal{N}$*, and an* x *adversary* $\mathcal{A}''$ *with complexity* $\mathcal{N}$*, such that*

$$\mathbf{Adv}^{\mathrm{x}}_{\text{Ascon-XOF}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{indiff}}_{\text{Ascon-XOF},\mathbf{Sim}}(\mathcal{A}') + \mathbf{Adv}^{\mathrm{x}}_{\$}(\mathcal{A}'') \,.$$

Here, $\mathbf{Adv}^{\mathrm{x}}_{\$}(\mathcal{A}'')$ *slightly abuses notation as* $\mathcal{A}''$ *gets direct access to* \$ *and aims to break* x *security.*

Note that $\mathbf{Adv}^{\mathrm{col}[\nu]}_{\$}(\mathcal{A}'') \leq \binom{\mathcal{N}}{2}/2^{\nu}$ and $\mathbf{Adv}^{\mathrm{pre}[\nu]}_{\$}(\mathcal{A}'') = \mathbf{Adv}^{\mathrm{sec}[\kappa,\,\nu]}_{\$}(\mathcal{A}'') = \mathcal{N}/2^{\nu}$.

8.3 Overview

Bertoni et al. [BDPV08] proved that the sponge construction is indifferentiable from a random oracle. In detail, they proved a bound up to

$$1 - \prod_{i=0}^{\mathcal{N}-1} \left(\frac{1 - \frac{i+1}{2^c}}{1 - \frac{i}{2^{r+c}}} \right) ,$$

which then gets approximated to $\binom{\mathcal{N}+1}{2}/2^c$. However, this approximation uses the inequality $1 - x \leq e^{-x}$ in two directions (first to lower bound, then to upper bound), de facto making this approximation a true approximation instead of a strict upper bound. An alternative approach would give a proper upper bound of the form $2\binom{\mathcal{N}+1}{2}/2^c$. We adopt their result but with this simplified upper bound.

Theorem 7 ([BDPV08]). *Let* $b, c, r, \mathcal{N} \in \mathbb{N}$ *with* $b = r + c$*. Consider the* Ascon-XOF *mode of Section 8.1 with parameters* b, c, r*. Let* $\mathcal{A}$ *be an adversary with complexity* $\mathcal{N}$*. There exists a simulator* **Sim** *with complexity* $\mathcal{O}(\mathcal{N})$ *queries such that*

$$\mathbf{Adv}^{\mathrm{indiff}}_{\text{Ascon-XOF},\mathbf{Sim}}(\mathcal{A}) \leq \frac{\mathcal{N}(\mathcal{N}+1)}{2^c} \,.$$

This result directly implies collision, preimage, and second preimage resistance using Lemma 3. However, Lefevre and Mennink [LM22] demonstrated that preimage resistance is in fact better than that, and we include their result.[12]

Theorem 8 ([BDPV08, LM22]). *Let* $b, c, r, \kappa, \nu \in \mathbb{N}$ *with* $b = r + c$*, and let* $\ell = \lceil \frac{\nu}{r} \rceil$*. Let* $\mathcal{N} \in \mathbb{N}$ *be such that, for the case of preimage resistance,* $\mathcal{N} \leq 2^{c-1}/3$ *and* $(\ell - 1)^2 \leq 2^b$*. Consider the* Ascon-XOF *mode of Section 8.1 with parameters* b, c, r*. Let* $\mathcal{A}$ *be an adversary with complexity* $\mathcal{N}$*. We have*

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{col}[\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) &\leq \frac{\mathcal{N}(\mathcal{N}+1)}{2^c} + \frac{\mathcal{N}(\mathcal{N}-1)}{2^{\nu+1}} \,, \\
\mathbf{Adv}^{\mathrm{pre}[\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) &\leq \min\left\{ \frac{\mathcal{N}(\mathcal{N}+1)}{2^c}, \frac{4\ell\mathcal{N}}{2^{\nu-r}} \right\} + \frac{4\mathcal{N}}{2^{\nu}} \,, \\
\mathbf{Adv}^{\mathrm{sec}[\kappa,\,\nu]}_{\text{Ascon-XOF}}(\mathcal{A}) &\leq \frac{\mathcal{N}(\mathcal{N}+1)}{2^c} + \frac{\mathcal{N}}{2^{\nu}} \,.
\end{aligned}$$

For the specific parameters of Ascon-Hash, Theorem 8 implies 128-bit collision and second preimage resistance but 192-bit preimage resistance. For Ascon-(C)XOF, Theorem 8

[12] In fact, the result of Lefevre and Mennink [LM22] is stronger in the sense that it applies to the construction underlying PHOTON [GPP11], which has a larger initial absorption and squeezing absorption rate. We consider their result in the context of Ascon-XOF.

implies $\min\{\nu/2, 128\}$-bit collision resistance, $\min\{\nu, 128\}$-bit second preimage resistance, and $\min\{\nu, 192\}$-bit preimage resistance, where ν is the minimal output size.

The bounds of Theorem 7 and Theorem 8 are tight, with matching attacks already given in the original specification [BDPV07, Section 5] and re-described in terminology matching Definition 10 by Lefevre and Mennink [LM22].

Proposition 10 ([BDPV07, LM22]). *Let $b, c, r, \kappa, \nu \in \mathbb{N}$ with $b = r + c$. Consider the* Ascon-XOF *mode of Section 8.1 with parameters b, c, r. There exists an adversary $\mathcal{A}$ with $\mathcal{N} \approx 2^{c/2}$, such that for any simulator* **Sim**,

$$\mathbf{Adv}^{\text{indiff}}_{\text{Ascon-XOF},\mathbf{Sim}}(\mathcal{A}) \approx 1\,.$$

In addition, there exist adversaries $\mathcal{A}_1^{\text{col}}$ with $\mathcal{N} \approx 2^{c/2}$, $\mathcal{A}_2^{\text{col}}$ with $\mathcal{N} \approx 2^{\nu/2}$, $\mathcal{A}_1^{\text{pre}}$ with $\mathcal{N} \approx \max\{2^{c/2}, 2^{\nu-r}\}$, $\mathcal{A}_2^{\text{pre}}$ with $\mathcal{N} \approx 2^{\nu}$, $\mathcal{A}_1^{\text{sec}}$ with $\mathcal{N} \approx 2^{c/2}$, and $\mathcal{A}_2^{\text{sec}}$ with $\mathcal{N} \approx 2^{\nu}$, such that

$$\begin{aligned}
&\mathbf{Adv}^{\text{col}[\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_1^{\text{col}}\right), \mathbf{Adv}^{\text{col}[\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_2^{\text{col}}\right) \approx 1\,,\\
&\mathbf{Adv}^{\text{pre}[\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_1^{\text{pre}}\right), \mathbf{Adv}^{\text{pre}[\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_2^{\text{pre}}\right) \approx 1\,,\\
&\mathbf{Adv}^{\text{sec}[\kappa,\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_1^{\text{sec}}\right), \mathbf{Adv}^{\text{sec}[\kappa,\nu]}_{\text{Ascon-XOF}}\left(\mathcal{A}_2^{\text{sec}}\right) \approx 1\,.
\end{aligned}$$

9 Ascon-PRF Mode and Its Security

We describe the mode underlying Ascon-PRF in Section 9.1, the security model in Section 9.2, and the security of the construction in Section 9.3.

9.1 Description of the Mode

The construction underlying Ascon-PRF is a tweaked version of the full-state keyed sponge [BDPV12, MRV15].[13] Let $b, c, r, c', r', k \in \mathbb{N}$ such that $b = r + c = r' + c'$, $k \leq b$, and let p be a cryptographic permutation over b bits. The function Ascon-PRF^p takes as input a key $K \in \{0,1\}^k$, a plaintext $P \in \{0,1\}^*$, and a length $l \in \mathbb{N}$. It returns a tag $T \in \{0,1\}^*$ with $|T| = l$. We have

$$\begin{aligned}
\text{Ascon-PRF}^p_K : \{0,1\}^* \times \mathbb{N} &\longrightarrow \{0,1\}^*\,,\\
(P, l) &\longrightarrow T \in \{0,1\}^l\,.
\end{aligned}$$

The most notable difference with the full-state keyed sponge is that Ascon-PRF features domain separation between the absorption phase and the squeezing phase. The construction is illustrated in Figure 5. Ascon-PRF is suggested to be instantiated with key size $k = 128$, absorption capacity and rate $(c, r) = (64, 256)$, and squeezing capacity and rate $(c', r') = (192, 128)$.

9.2 Security

We consider plain multi-user PRF security for Ascon-PRF.

Definition 11. Consider the Ascon-PRF mode of Section 9.1. Let $(\$_m)_{m=1}^{\mu}$ be a family of μ independent random functions, $p \xleftarrow{\$} \text{Perm}(b)$, and $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. The PRF security of Ascon-PRF against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-prf}}_{\text{Ascon-PRF}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\text{Ascon-PRF}^p_{K_m}\right)_{m=1}^{\mu}, p^{\pm}\,;\, (\$_m)_{m=1}^{\mu}, p^{\pm}\right)\,.$$

[13] The Ascon-PRF specification also comes with a small-input variant Ascon-PRFshort, that is basically a truncated permutation construction with key blinding.

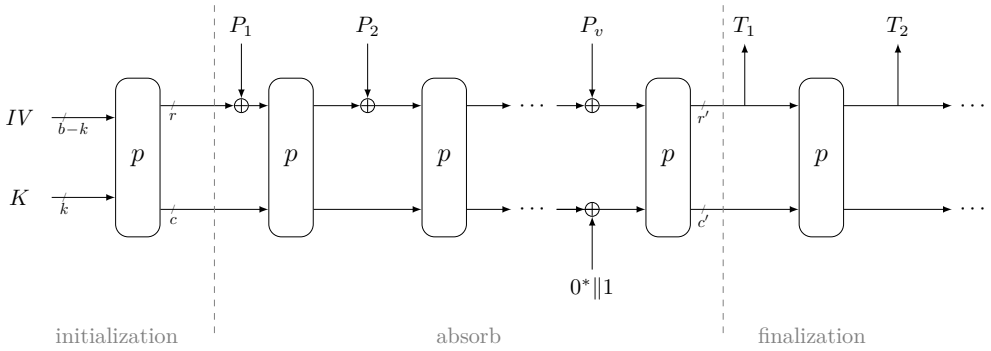

Figure 5: The Ascon-PRF mode of operation. Here, P is injectively padded as $(P_1, \ldots, P_v) \leftarrow \mathrm{pad}_r^{10^*}(P)$.

9.3 Overview

Ascon-PRF is basically a full-state keyed sponge [BDPV12]. Mennink et al. [MRV15] gave an analysis of the full-state keyed sponge, but it included a so-called proof-inherent "multiplicity" term. Daemen et al. [DMV17] derived an analysis of the full-state keyed duplex, and their bound also applies to Ascon-PRF. However, in our terminology, their bound has a term $\mathcal{MN}/2^c$, but the attack matching this term is actually thwarted by the domain separation between absorption and squeezing in Ascon-PRF. Because of this, Mennink [Men23] dived into the existing duplex proofs [DMV17, DM19b] and improved them to obtain a bound specifically tailored to Ascon-PRF. However, with respect to multicollision handling, their proofs adopt the first strategy of Section 3.4, whereas the analysis of Ascon-AE of Sections 4 and 5 follows the second strategy. We thus take the bound of Mennink, but adapt it by taking the second multicollision strategy. (This boils down to replacing the fraction $2\theta(\mathcal{N}+1)/2^c$ by $2\theta\mathcal{N}/2^c$, though for a different meaning of the value θ.)

Theorem 9 ([Men23]). *Let $b, c, r, c', r', k \in \mathbb{N}$ with $b = r + c = r' + c'$ and $k \leq b$. Let $\mathcal{N}, \mathcal{M}, Q \in \mathbb{N}$. Consider the* Ascon-PRF *mode of Section 9.1 with parameters b, c, r, c', r', k. Let $\mathcal{A}$ be an adversary with complexity $(\mathcal{N}, Q, \mathcal{M})$. We have*

$$\mathbf{Adv}^{\mu\text{-prf}}_{\text{Ascon-PRF}}(\mathcal{A}) \leq \frac{2\mathrm{mucol}\left(\mathcal{M}, 2^{r'}\right)\mathcal{N}}{2^{c'}} + \frac{(\mathcal{M}-Q)Q}{2^b - Q} + \frac{2\binom{\mathcal{M}}{2}}{2^b} + \frac{Q(\mathcal{M}-Q)}{2^{\min\{c'+k,b\}}} + \frac{\mu\mathcal{N}}{2^k} + \frac{\binom{\mu}{2}}{2^k}\,.$$

Using that $\mu \leq \mathcal{M} \ll \mathcal{N}$ (cf., Section 3.3) and $\mathrm{mucol}(q, R) = \mathcal{O}\left(1 + \frac{q}{R}\right)$, we obtain a bound of the order

$$\mathbf{Adv}^{\mu\text{-prf}}_{\text{Ascon-PRF}}(\mathcal{A}) = \mathcal{O}\left(\frac{\mu\mathcal{N}}{2^k} + \frac{\mathcal{N}}{2^{c'}} + \frac{\mathcal{MN}}{2^b}\right). \tag{36}$$

The bound (36) is tight. The same type of attacks as those in Proposition 1 apply here, as we show in Proposition 11.

Proposition 11. *Let $b, c, r, c', r', k \in \mathbb{N}$ with $b = r + c = r' + c'$ and $k \leq c$. Consider the* Ascon-PRF *mode of Section 9.1 with parameters b, c, r, c', r', k. There exist adversaries $\mathcal{A}_1$ with $\mathcal{N} \approx 2^k/\mu$, and $\mathcal{A}_2$ with $(\mathcal{M} + 2^{r'})\mathcal{N} \approx 2^b$, such that*

$$\mathbf{Adv}^{\mu\text{-prf}}_{\text{Ascon-PRF}}(\mathcal{A}_1)\,,\ \mathbf{Adv}^{\mu\text{-prf}}_{\text{Ascon-PRF}}(\mathcal{A}_2) \approx 1\,.$$

Proof. The adversary $\mathcal{A}_1$ acts similarly to one described in Section 7.1.2: it tries to guess one of the user's keys by doing permutation queries, and checks them against the outputs

of each of the μ Ascon-PRF instances to determine whether one of its guesses is correct. Similarly, adversary $\mathcal{A}_2$ follows the strategy from Section 7.1.3, as it aims to guess an internal state. This internal state must be during the squeezing phase, and successfully guessing it directly implies key recovery. □

10 Conclusion

We presented a general discussion of existing, and new, results on the generic security of the Ascon-AE authenticated encryption scheme, as well as the Ascon-Hash/Ascon-(C)XOF hash/(C)XOF function, and the Ascon-PRF pseudorandom function. On the way, we observed that, even though state-of-the-art appeared quite broad at first sight, there were still many gaps to be filled. In particular, some proofs had to be revisited in light of the assumption (in our work) that the outer and inner permutation of Ascon-AE are identical (cf., Remark 1), but also some existing results were not entirely correct/accurate, and results in bounded leakage resilience and release of unverified plaintext were lacking.

10.1 What We Did Not Cover

Even though our treatment is rather broad, we do not cover *all* possible security models. We elaborate on three directions that we did not cover:

- We did not cover related-key security. We do, in fact, consider multi-user security, where the adversary has access to μ instances of the scheme, but we assume independence of these μ keys. If we would have stretched the analysis to arbitrary distributions, e.g., as in Daemen et al. [DMV17, Section 2.1], this would imply *some forms of* related-key security, but this would significantly add to the complexity of the proofs;
- While we cover leakage resilience in Section 5.1, we do not cover security under fault attacks, where the adversary may introduce faults in the implementation and that way retrieve secret information. Fruitful starting points for the analysis of Ascon-AE in this setting are [DMP22, SKP22, BGP+23];
- There has been significant recent interest in committing security of authenticated encryption schemes, where the adversary has freedom in choosing the keys [GLR17, LGR21, ADG+22, CR22, BH22]. Clearly, there is a relation between hash function security and key committing security of authenticated encryption schemes, but the blinding of the keys in Ascon-AE makes this relation not directly applicable. Naito et al. [NSS23] proved key committing security of the Ascon-AE mode with zero-padding, where security (in part) depends on the size of the zero-padding. An alternative security proof up to a comparable bound is given by Krämer et al. [KSW24].

In addition, our analysis does not cover variations to the Ascon-AE scheme of Section 2:

- The draft standard published by NIST [SMKK24] also specifies an option to implement Ascon-AE with nonce masking, where the nonce gets blinded with additional key material, following Dobraunig and Mennink [DM24]. The goal of this mechanism is to enhance the multi-user security of the mode, i.e., to replace the term $\frac{\mu\mathcal{N}}{2^k}$ with $\frac{\mathcal{N}}{2^k}$. We did not cover this mechanism in our proofs, but remark that the change only affects the initialization, that Dobraunig and Mennink proposed their mechanisms as extension to the original duplex proofs [DMV17, DM19a], and that our results may similarly be generalized by an isolated analysis;
- The encryption in duplex-based authenticated encryption, and Ascon-AE in particular, consists of bitwise addition of plaintext into the outer part, but upon decryption,

the adversary chooses the ciphertext and has free choice in selecting the r outer bits of the state. This gives the adversary additional power, especially in settings such as release of unverified plaintext (Section 5.3). One way to mitigate this decrease in security is to apply a more advanced mechanism to absorb plaintext and squeeze ciphertext, e.g., following the approach of Beetle [CDNY18]. A general analysis of security of Beetle-style authenticated encryption was given by Chakraborty et al. [CJN20]. However, such mechanism cannot be implemented with Ascon-AE as a black-box (unlike above variation); instead, it would basically be a different scheme.

10.2 Future Research

Finally, we wish to point out two directions where our models and assumptions limit, and where further research on Ascon-AE would be worthwhile:

- For leakage resilience (Section 5.1), we started from the notion of nonce-misuse resilience, and we additionally covered any possible leakage function that is independent of the permutation p. These two modeling decisions, together, implied that limited and unlimited leakage are equivalent. However, while the first modeling decision is rather fair, the second modeling decision is fairly liberal as in practice, an adversary cannot freely choose the leakage function. Therefore, *for specific leakage functions*, we expect much better security bounds. For example, if we limit the leakage function to leak the Hamming weight of the first byte, the security bound may be closer to security with no leakage than security with unlimited leakage. However, performing security analysis for more complex leakage functions is very subtle and technical. For example, Berendsen and Mennink [BM24] recently considered the security of the suffix keyed sponge under Hamming weight leakage, and it seems not trivial to generalize our analysis (of Theorem 4) to Hamming weight leakage, nor to generalize the analysis of [BM24] on the suffix keyed sponge to Ascon-AE;
- We can be reasonably confident that the Ascon permutation (and, in the context of Ascon-AE, the outer permutation p_{o} in particular) is sufficiently strong. That said, they are definitely not perfectly random, and the more one considers round-reduced variants (e.g., the inner permutation p_{i} versus the outer permutation p_{o}), the more one diverges from this assumption. Harshly said, our results do not apply to the actual Ascon schemes. However, they do give some certainty, namely that no generic attacks are possible beyond the proven bounds, and also state-recovery security says that some level of security is still achieved even if one of the inner states leaks. Having said that, it is a very interesting research question to try to prove security of Ascon under weaker assumption on the permutation.

Acknowledgements. We want to thank the Ascon team for feedback on this work, as well as the anonymous reviewers for their valuable comments. Charlotte Lefevre is supported by the Netherlands Organisation for Scientific Research (NWO) under grant OCENW.KLEIN.435. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

[ABD+23] Roberto Avanzi, Subhadeep Banik, Orr Dunkelman, Maria Eichlseder, Shibam Ghosh, Marcel Nageler, and Francesco Regazzoni. The QARMAv2 Family of Tweakable Block Ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(3):25–73, 2023.

[ABL+13] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and Authenticated Online Ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 424–443. Springer, 2013.

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to Securely Release Unverified Plaintext in Authenticated Encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 105–125. Springer, 2014.

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to Abuse and Fix Authenticated Encryption Without Key Commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3291–3308. USENIX Association, 2022.

[ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness with Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2017.

[ADMV15] Elena Andreeva, Joan Daemen, Bart Mennink, and Gilles Van Assche. Security of Keyed Sponge Constructions Using a Modular Proof Approach. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 364–384. Springer, 2015.

[AHMN10] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A Lightweight Hash. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2010.

[AJN14] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. NORX v1. Submission to CAESAR competition, 2014.

[AMP10] Elena Andreeva, Bart Mennink, and Bart Preneel. Security Reductions of the Second Round SHA-3 Candidates. In Mike Burmester, Gene Tsudik, Spyros S. Magliveras, and Ivana Ilic, editors, *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, volume 6531 of *Lecture Notes in Computer Science*, pages 39–53. Springer, 2010.

[Ava17] Roberto Avanzi. The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Con-

structions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge Functions. Ecrypt Hash Workshop 2007, May 2007.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BDPV11a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[BDPV11b] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Security of the Keyed Sponge Construction. Symmetric Key Encryption Workshop, February 2011.

[BDPV12] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers, July 2012.

[BGP+20] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. TEDT, a Leakage-Resist AEAD Mode for High Physical Security Applications. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):256–320, 2020.

[BGP+23] Francesco Berti, Chun Guo, Thomas Peters, Yaobin Shen, and François-Xavier Standaert. Secure Message Authentication in the Presence of Leakage and Faults. *IACR Trans. Symmetric Cryptol.*, 2023(1):288–315, 2023.

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient Schemes for Committing Authenticated Encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 845–875. Springer, 2022.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The

SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[BKL+11] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. spongent: A Lightweight Hash Function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 1998.

[BM24] Henk Berendsen and Bart Mennink. Tightening Leakage Resilience of the Suffix Keyed Sponge. *IACR Trans. Symmetric Cryptol.*, 2024(1):459–496, 2024.

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated Encryption in the Face of Protocol and Side Channel Leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 693–723. Springer, 2017.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *J. Cryptol.*, 21(4):469–491, 2008.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems*

- *CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.

[BR06] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org/2013/404.

[CAE14] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, May 2014. http://competitions.cr.yp.to/caesar.html.

[CDD+19] Donghoon Chang, Nilanjan Datta, Avijit Dutta, Bart Mennink, Mridul Nandi, Somitra Sanadhya, and Ferdinand Sibleyras. Release of Unverified Plaintext: Tight Unified Model and Application to ANYDAE. *IACR Trans. Symmetric Cryptol.*, 2019(4):119–146, 2019.

[CDH+12] Donghoon Chang, Morris Dworkin, Seokhie Hong, John Kelsey, and Mridul Nandi. A Keyed Sponge Construction with Pseudorandomness in the Standard Model. NIST SHA–3 Workshop, March 2012.

[CDK09] Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.

[CDN23] Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Exact Security Analysis of ASCON. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 346–369. Springer, 2023.

[CDN24] Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Tight Multi-user Security of Ascon and Its Large Key Extension. In Tianqing Zhu and Yannan Li, editors, *Information Security and Privacy - 29th Australasian Conference, ACISP 2024, Sydney, NSW, Australia, July 15-17, 2024, Proceedings, Part I*, volume 14895 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2024.

[CDNY18] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle Family of Lightweight and Secure Authenticated Encryption Ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[CJN20] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the Security of Sponge-type Authenticated Encryption Modes. *IACR Trans. Symmetric Cryptol.*, 2020(2):93–119, 2020.

[CLL19] Wonseok Choi, ByeongHak Lee, and Jooyoung Lee. Indifferentiability of Truncated Random Permutations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 175–195. Springer, 2019.

[CR22] John Chan and Phillip Rogaway. On Committing Authenticated-Encryption. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26-30, 2022, Proceedings, Part II*, volume 13555 of *Lecture Notes in Computer Science*, pages 275–294. Springer, 2022.

[CS14] Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, 2014.

[DDNT23] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Tight Multi-User Security Bound of DbHtS. *IACR Trans. Symmetric Cryptol.*, 2023(1):192–223, 2023.

[DEMS14] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1. Submission to CAESAR competition, 2014.

[DEMS19] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. Winning Submission to NIST Lightweight Cryptography, 2019.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *J. Cryptol.*, 34(3):33, 2021.

[DEMS24] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon MAC, PRF, and Short-Input PRF - Lightweight, Fast, and Efficient Pseudorandom Functions. In Elisabeth Oswald, editor, *Topics in Cryptology - CT-RSA 2024 - Cryptographers' Track at the RSA Conference 2024, San Francisco, CA, USA, May 6-9, 2024, Proceedings*, volume 14643 of *Lecture Notes in Computer Science*, pages 381–403. Springer, 2024.

[DHT17] Wei Dai, Viet Tung Hoang, and Stefano Tessaro. Information-Theoretic Indistinguishability via the Chi-Squared Method. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 497–523. Springer, 2017.

[Din24] Itai Dinur. Tight Indistinguishability Bounds for the XOR of Independent Random Permutations by Fourier Analysis. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2024.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 621–630. ACM, 2009.

[DM19a] Christoph Dobraunig and Bart Mennink. Leakage Resilience of the Duplex Construction. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 225–255. Springer, 2019.

[DM19b] Christoph Dobraunig and Bart Mennink. Security of the Suffix Keyed Sponge. *IACR Trans. Symmetric Cryptol.*, 2019(4):223–248, 2019.

[DM20] Christoph Dobraunig and Bart Mennink. Tightness of the Suffix Keyed Sponge Bound. *IACR Trans. Symmetric Cryptol.*, 2020(4):195–212, 2020.

[DM24] Christoph Dobraunig and Bart Mennink. Generalized Initialization of the Duplex Construction. In Christina Pöpper and Lejla Batina, editors, *Applied Cryptography and Network Security - 22nd International Conference, ACNS 2024, Abu Dhabi, United Arab Emirates, March 5-8, 2024, Proceedings, Part II*, volume 14584 of *Lecture Notes in Computer Science*, pages 460–484. Springer, 2024.

[DMMS21] Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring Crypto-Physical Dark Matter and Learning with Physical Rounding Towards Secure and Efficient Fresh Re-Keying. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):373–401, 2021.

[DMP22] Christoph Dobraunig, Bart Mennink, and Robert Primas. Leakage and Tamper Resilient Permutation-Based Cryptography. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 859–873. ACM, 2022.

[DMV17] Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-State Keyed Duplex with Built-In Multi-user Support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 606–637. Springer, 2017.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-Resilient Cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 293–302. IEEE Computer Society, 2008.

[DP10] Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.

[DPVR00] Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie Proposal: NOEKEON. First Open NESSIE Workshop, 2000.

[FFL12] Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 196–215. Springer, 2012.

[FH15] Benjamin Fuller and Ariel Hamlin. Unifying Leakage Classes: Simulatable Leakage and Pseudoentropy. In Anja Lehmann and Stefan Wolf, editors, *Information Theoretic Security - 8th International Conference, ICITS 2015, Lugano, Switzerland, May 2-5, 2015. Proceedings*, volume 9063 of *Lecture Notes in Computer Science*, pages 69–86. Springer, 2015.

[FPS12] Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical Leakage-Resilient Symmetric Cryptography. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 2012.

[GBKR23] Henri Gilbert, Rachelle Heim Boissier, Louiza Khati, and Yann Rotella. Generic Attack on Duplex-Based AEAD Modes Using Random Function Statistics. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 348–378. Springer, 2023.

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message Franking via Committing Authenticated Encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 66–97. Springer, 2017.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON Family of Lightweight Hash Functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.

[GPPS19a] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Authenticated Encryption with Nonce Misuse and Physical Leakage: Definitions, Separation Results and First Construction - (Extended Abstract). In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4,*

2019, Proceedings, volume 11774 of *Lecture Notes in Computer Science*, pages 150–172. Springer, 2019.

[GPPS19b] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. Cryptology ePrint Archive, Report 2019/193, 2019. http://eprint.iacr.org/2019/193 (full version of [GPPS20]).

[GPPS20] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. *IACR Trans. Symmetric Cryptol.*, 2020(1):6–42, 2020.

[GPT15] Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. The Exact PRF Security of Truncation: Tight Bounds for Keyed Sponges and Truncated CBC. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 368–387. Springer, 2015.

[GTW24] Felix Günther, Martin Thomson, and Christopher A. Wood. Usage Limits on AEAD Algorithms. Internet Engineering Task Force, Internet-Draft, draft-irtf-cfrg-aead-limits-09, October 2024.

[HRRV15] Viet Tung Hoang, Reza Reyhanitabar, Phillip Rogaway, and Damian Vizár. Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 493–517. Springer, 2015.

[HSH+06] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.

[JLM14] Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 85–104. Springer, 2014.

[JLM+19] Philipp Jovanovic, Atul Luykx, Bart Mennink, Yu Sasaki, and Kan Yasuda. Beyond Conventional Security in Sponge-Based Authenticated Encryption Modes. *J. Cryptol.*, 32(3):895–940, 2019.

[JN20] Ashwin Jha and Mridul Nandi. Tight Security of Cascaded LRW2. *J. Cryptol.*, 33(3):1272–1317, 2020.

[KR19] Yael Tauman Kalai and Leonid Reyzin. A Survey of Leakage-Resilient Cryptography. Cryptology ePrint Archive, Report 2019/302, 2019. http://eprint.iacr.org/2019/302.

[KSW24] Juliane Krämer, Patrick Struck, and Maximiliane Weishäupl. Committing AE from Sponges Security Analysis of the NIST LWC Finalists. *IACR Trans. Symmetric Cryptol.*, 2024(4):191–248, 2024.

[Lef24] Charlotte Lefevre. A Note on Adversarial Online Complexity in Security Proofs of Duplex-Based Authenticated Encryption Modes. Cryptology ePrint Archive, Report 2024/213, 2024. http://eprint.iacr.org/2024/213.

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning Oracle Attacks. In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 195–212. USENIX Association, 2021.

[LM22] Charlotte Lefevre and Bart Mennink. Tight Preimage Resistance of the Sponge Construction. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 185–204. Springer, 2022.

[LM24] Charlotte Lefevre and Bart Mennink. Generic Security of the Ascon Mode: On the Power of Key Blinding. In Maria Eichlseder and Sébastien Gambs, editors, *Selected Areas in Cryptography, 31st International Workshop, SAC 2024, Montréal, Quebec, Canada, August 26-27, Revised Selected Papers*, Lecture Notes in Computer Science. Springer, 2024. to appear.

[LMO+14] Jake Longo, Daniel P. Martin, Elisabeth Oswald, Daniel Page, Martijn Stam, and Michael Tunstall. Simulatable Leakage: Analysis, Pitfalls, and New Constructions. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 223–242. Springer, 2014.

[LMP17] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing Multi-key Security Degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017.

[LNS18] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic Attacks Against Beyond-Birthday-Bound MACs. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 306–336. Springer, 2018.

[Luc00] Stefan Lucks. The Sum of PRPs Is a Secure PRF. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 2000.

[Men18] Bart Mennink. Key Prediction Security of Keyed Sponges. *IACR Trans. Symmetric Cryptol.*, 2018(4):128–149, 2018.

[Men23] Bart Mennink. Understanding the Duplex and Its Security. *IACR Trans. Symmetric Cryptol.*, 2023(2):1–46, 2023.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[MRV15] Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 465–489. Springer, 2015.

[Nat15] National Institute of Standards and Technology. FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.

[NIS07] NIST. SHA-3 Project, February 2007. https://csrc.nist.gov/projects/hash-functions/sha-3-project.

[NIS19] NIST. Lightweight Cryptography, February 2019. https://csrc.nist.gov/Projects/Lightweight-Cryptography.

[NRS14] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2014.

[NSS23] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Committing Security of Ascon: Cryptanalysis on Primitive and Proof on Mode. *IACR Trans. Symmetric Cryptol.*, 2023(4):420–451, 2023.

[NY16] Yusuke Naito and Kan Yasuda. New Bounds for Keyed Sponges with Extendable Output: Independence Between Capacity and Message Length. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2016.

[Pat91] Jacques Patarin. *Étude des Générateurs de Permutations Basés sur le Schéma du D.E.S.* PhD thesis, Université Paris 6, Paris, France, November 1991.

[Pat08a] Jacques Patarin. A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations. In Reihaneh Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 232–248. Springer, 2008.

[Pat08b] Jacques Patarin. The "Coefficients H" Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.

[PDM+18] Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, and Jörg Schwenk. Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels. In William Enck and Adrienne Porter Felt, editors, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 549–566. USENIX Association, 2018.

[Pie09] Krzysztof Pietrzak. A Leakage-Resilient Mode of Operation. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 462–482. Springer, 2009.

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 96–108. ACM, 2015.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.

[RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.

[Shr04] Tom Shrimpton. A Characterization of Authenticated-Encryption as a Form of Chosen-Ciphertext Security. Cryptology ePrint Archive, Report 2004/272, 2004. http://eprint.iacr.org/2004/272.

[SKP22] Sayandeep Saha, Mustafa Khairallah, and Thomas Peyrin. Exploring Integrity of AEADs with Faults: Definitions and Constructions. *IACR Trans. Symmetric Cryptol.*, 2022(4):291–324, 2022.

[SMKK24] Meltem Sönmez Turan, Kerry A. McKay, Jinkeon Kang, and John Kelsey. Ascon-Based Lightweight Cryptography Standards for Constrained Devices. NIST SP 800-232 ipd, November 2024. https://csrc.nist.gov/pubs/sp/800/232/ipd.

[SPY+10] François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage Resilient Cryptography in Practice.

In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security - Foundations and Practice*, Information Security and Cryptography, pages 99–134. Springer, 2010.

[SPY13] François-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-Resilient Symmetric Cryptography under Empirically Verifiable Assumptions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 335–352. Springer, 2013.

[YSPY10] Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 141–151. ACM, 2010.